Robust and Multi-Modal Analysis of Traffic and People

Bahnsen, Chris Holmberg

DOI (link to publication from Publisher):
10.54337/aau308456532

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

ROBUST AND MULTI-MODAL ANALYSIS OF TRAFFIC AND PEOPLE

BY
CHRIS HOLMBERG BAHNSEN

DISSERTATION SUBMITTED 2019

AALBORG UNIVERSITY
DENMARK

Robust and Multi-Modal Analysis of Traffic and People

Ph.D. Dissertation

Chris Holmberg Bahnsen

Dissertation submitted March 13, 2019

Dissertation submitted:	March 2019
PhD supervisor:	Professor Thomas B. Moeslund Aalborg University
PhD committee:	Professor Lazaros Nalpantidis (chairman) Aalborg University
	Professor Henrik Karstoft Aarhus University
	Professor Michael Felsberg Linköping University
PhD Series:	Technical Faculty of IT and Design, Aalborg University
Department:	Department of Architecture, Design and Media Technology

ISSN (online): 2446-1628
ISBN (online): 978-87-7210-358-7

Published by:
Aalborg University Press
Langagervej 2
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

Printed in Denmark by Rosendahls, 2019

Curriculum Vitae

Chris Holmberg Bahnsen

Chris Holmberg Bahnsen received the B.Sc. degree in electrical engineering and the M.Sc. degree in vision, graphics, and interactive systems from Aalborg University, Denmark, in 2011 and 2013, respectively. He worked as a research assistant before starting his PhD in July 2014 with the Visual Analysis of People Laboratory at the Section of Media Technology, Aalborg University.

He has been a Visiting Scholar for three months at the Advanced Driver Assistance Systems (ADAS) research group located at the Computer Vision Center, Universitat Autònoma de Barcelona, Spain.

His main interests include computer vision and machine learning, particularly in the area of road traffic surveillance. He has been involved in supervision and teaching of undergraduate and graduate students within image processing and computer vision.

Curriculum Vitae

Abstract

This PhD covers work conducted from 2013 to 2018 within two overall themes: multi-modal analysis and robust traffic analysis

Within multi-modal analysis, we investigated how to obtain synchronized and registered imagery from visual, depth, and thermal sensors. We used our tri-modal acquisition and registration platform to conduct research within people re-identification and people segmentation. As part of this work, we released a publicly available tri-modal dataset for people segmentation. We developed and extended an annotation toolbox that has subsequently been used for many other research projects in our laboratory. In traffic surveillance, we have investigated how to combine information from visual and thermal cameras using the related contextual information.

In the work within robust traffic analysis, we have studied the influence of rainfall and snowfall on visual traffic surveillance. We surveyed the field of rain removal algorithms and selected six algorithms to investigate if the removal of rain from traffic surveillance video would increase the performance of subsequent background subtraction, instance segmentation, and feature tracking algorithms. We collected and annotated a new publicly available dataset consisting of visual-thermal traffic surveillance video.

Furthermore, we have investigated the use of fully synthetic sequences from a virtual world where rain is rendered in the entirety of the scene. On the basis hereof, we trained a rain removal algorithm.

Traffic researchers are analyzing the behavior of road users in order to understand the causation of accidents and improve road safety. To reduce the amount of video for manual inspection by the traffic researchers, we have developed the RUBA software tool. We have tested RUBA for detection and counting of turning road user movements at urban intersections and compared it against a more advanced, feature-based tracker. In extension of our work within traffic surveillance, we investigated several options for establishing a portable video acquisition platform, which led to the construction of a new portable pole.

As part of the dissemination activities, we have authored an article on deep learning for the Danish magazine on popular science, *Aktuel Naturvidenskab*.

Abstract

Resumé

Denne PhD-afhandling afdækker arbejder fra 2013 til 2018 indenfor to overordnede temaer: multi-modal analyse og robust trafikanalyse.

Indenfor multi-modal analyse har vi undersøgt, hvordan man optager synkroniserede og registrerede billeder fra visuelle, dybde-baserede og termiske sensorer. Vi har brugt vores tri-modale optage- og registreringsplatform til at foretage forskning indenfor re-identifikation og segmentering af personer. Som en del af dette arbejde har vi udgivet et offentligt tilgængeligt tri-modalt dataset til personsegmentering. Vi har udviklet og udbygget et annoteringsværktøj, som sidenhen er blevet benyttet til mange andre forskningsprojekter i vores laboratorie. Indenfor trafikovervågning har vi undersøgt, hvordan man kombinerer information fra visuelle og termiske kameraer ved hjælp af kontekstuel information.

Vi har i vores arbejde med robust trafikanalyse studeret regn- og snevejrs påvirkning på visuel trafikovervågning. Vi har undersøgt området indenfor regnfjernelses-algoritmer og udvalgt seks af algoritmerne til at undersøge, om fjernelse af regn fra trafikovervågningsvideoer vil øge ydeevnen af efterfølgende baggrundssubtraktion-, instanssegmentering- og feature-tracking-algoritmer. Vi indsamlede og annoterede et nyt datasæt indeholdende visuel-termisk trafikovervågningsvideo. Derudover har vi undersøgt brugen af fuldt syntetiske sekvenser fra en virtuel verden, hvor regn er renderet i hele scenen. På baggrund af disse sekvenser har vi trænet en regnfjernelsesalgoritme.

Trafikforskere analyserer trafikanternes adfærd for at forstå ulykkesårsager og forbedre trafiksikkerheden. For at reducere mængden af video, som trafikforskerne manuelt skal analysere, har vi udviklet softwareværktøjet RUBA. Vi har testet RUBA til detektion og optælling af drejende trafikanter ved bynære trafikkryds og sammenlignet det med en mere avanceret, feature-baseret tracker. I forlængelse af vores arbejde indenfor trafikovervågning har vi undersøgt flere forskellige muligheder for at etablere en portabel videooptagelsesplatform, hvilket førte til konstruktion af en ny portabel mast.

Som en del af formidlingsaktiviteterne har vi forfattet en artikel om deep learning til det danske populærvidenskabelige magasin, *Aktuel Naturvidenskab*.

Contents

Thesis Details

Thesis Title	Robust and Multi-Modal Analysis of Traffic and People
Ph.D. Student	Chris Holmberg Bahnsen
Supervisor	Professor Thomas B. Moeslund, Aalborg University

The main body of this thesis consists of the following papers and technical reports:

Multi-Modal Analysis

[A] Andreas Mogelmose, Chris Bahnsen, Thomas Moeslund, Albert Clapes, and Sergio Escalera, *Tri-Modal Person Re-Identification with RGB, Depth and Thermal Features*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2013, pp. 301–307.

[B] Cristina Palmero, Albert Clapés, Chris Bahnsen, Andreas Møgelmose, Thomas B Moeslund, and Sergio Escalera, *Multi-Modal RGB–Depth–Thermal Human Body Segmentation*, International Journal of Computer Vision **118** (2016), no. 2, 217–239.

[C] Andreas Møgelmose, Chris Bahnsen, and Thomas B Moeslund, *Comparison of Multi-Shot Models for Short-Term Re-Identification of People Using RGB-D Sensors.*, Proceedings of the 10th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP), 2015, pp. 244–251.

[D] Chris H. Bahnsen, Andreas Møgelmose, and Thomas B. Moeslund, *The AAU Multimodal Annotation Toolboxes: Annotating Objects in Images and Videos*, arXiv preprint arXiv:1809.03171 (2018).

[E] Thiemo Alldieck, Chris H Bahnsen, and Thomas B Moeslund, *Context-Aware Fusion of RGB and Thermal Imagery for Traffic Monitoring*, Sensors **16** (2016), no. 11, 1947.

Robust Traffic Analysis

[F] Chris H. Bahnsen and Thomas B. Moeslund, *Rain Removal in Traffic Surveillance: Does it Matter?*, IEEE Transactions on Intelligent Transportation Systems (Early Access), 2018, pp. 1–18

[G] Chris H. Bahnsen, David Vázquez, Antonio M. López, and Thomas B. Moeslund, *Learning to Remove Rain in Traffic Surveillance by Using Synthetic Data*, Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP), in press, 2019.

[H] Chris Bahnsen and Thomas B Moeslund, *Detecting Road User Actions in Traffic Intersections Using RGB and Thermal Video*, Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on, IEEE, 2015, pp. 1–6.

[I] Chris Bahnsen and Thomas B Moeslund, *Detecting Road Users at Intersections Through Changing Weather Using RGB-Thermal Video*, International Symposium on Visual Computing, Springer, 2015, pp. 741–751.

[J] Tanja K O Madsen, Chris Bahnsen, Harry Lahrmann, and Thomas B Moeslund, *Automatic Detection of Conflicts at Signalized Intersections*, Workshop on the Comparison of Surrogate Measures of Safety Extracted from Video Data, Transportation Research Board 93rd Annual Meeting, 2014.

[K] Niels Agerholm, Charlotte Tønning, Tanja Kidholm Osmann Madsen, Chris Holmberg Bahnsen, Thomas B Moeslund, and Harry Spaabæk Lahrmann, *Road User Behaviour Analyses Based on Video Detections: Status and Best Practice Examples from the RUBA Software*, 24th ITS World Congress Montreal 2017, pp. 1–10.

[L] Morten B. Jensen, Chris H. Bahnsen, Harry S. Lahrmann, Tanja K. O. Madsen, and Thomas B. Moeslund, *Collecting Traffic Video Data using Portable Poles: Survey, Proposal, and Analysis*, Journal of Transportation Technologies, 2018, Vol. 8 No. 4, pp. 376–400

[M] Chris H. Bahnsen, Tanja K. O. Madsen, Morten Bornø Jensen, Harry Lahrmann, and Thomas B Moeslund, *The RUBA Watchdog Video Analysis Tool*, Deliverable submitted within the InDeV EU project, 2018. Based on the public wiki page at `https://bitbucket.org/aauvap/ruba/wiki/`.

Dissemination Activities

[N] Morten Bornø Jensen, Chris H. Bahnsen, Kamal Nasrollahi, and Thomas B. Moeslund, *Deep Learning: Et Gennembrud inden for Kunstig Intelligens*, Aktuel Naturvidenskab **2** (2018), 8–13.

This thesis has been submitted for assessment in partial fulfilment of the PhD degree. The thesis is based on the submitted or published scientific papers which are listed above. Parts of the papers are used directly or indirectly in the extended summary of the thesis. As part of the assessment, co-author statements have been made available to the assessment committee and are also available at the Faculty.

Preface

This thesis is submitted as a collection of papers in partial fulfillment of a PhD study at the Section of Media Technology, Aalborg University, Denmark. The papers are divided into two fields, multi-modal analysis and robust traffic analysis. Part one of the thesis contains an overview of the state-of-the-art in these fields and the contributions which have been made to them during this work. This is followed by one part containing selected papers published as part of this PhD in each of the two fields. The final part covers the dissemination activities.

This project has been carried out from 2013-2018, mainly in the Visual Analysis of People Laboratory at Aalborg University, but with a research stay at the Advanced Driver Assistance Systems (ADAS) research group at the Computer Vision Center, Universitat Autònoma de Barcelona, Spain.

I would like to thank my supervisor, Professor Thomas B. Moeslund, for encouraging me to pursue the PhD as well as providing good supervision throughout the process. I am thankful for the discussions and conversations, both on-topic and off-topic, that I have had with my colleagues at the Visual Analysis of People Laboratory, especially Rikke Gade, Morten Bornø Jensen, Andreas Møgelmose, and Anders Jørgensen. Thanks to Professor Antonio M. Lopez for hosting me at the Computer Vision Center in Barcelona, to David Vázquez who opened my eyes to the brave new world of deep learning, and to Gemma Rotger and Dena Bazazian for friendly conversations in Barcelona. I would also like to thank my colleagues at the Traffic Safety Research Group at Aalborg University for excellent cross-disciplinary collaboration throughout the period.

I am especially thankful for my wife Camilla and our daughter Johanne for your unconditional love, support, and understanding during approaching deadlines. Regardless of how the day at work has went, I have been grateful to return home to your open arms.

Chris Holmberg Bahnsen
Aalborg University, March 13, 2019

Part I

Overview of the work

Chapter 1

Introduction

Detection, classification, and tracking of one or more physical objects is an ability that we as humans take for granted. Using our visual cortex, we have learned to interpret the world as seen through our eyes. We do this subconsciously without further thought and effortlessly integrate the visual sensations with our abilities to hear, feel, and smell.

The field of *computer vision* is based on the desire to equip machines with similar capabilities of observing the world. First thought of as a summer project in artificial intelligence [3], computer vision has emerged as an important field in its own.

Automated analysis of the visual world by computer vision algorithms enables numerous purposes such as self-driving cars, industrial quality control, and automated surveillance of people or property. Combined with appropriate hardware and application-specific tailoring, computer vision algorithms will replace human workers in mundane, repetitive tasks and enable these workers to focus on higher-value, more creative and flexible tasks. The deployment of computer vision algorithms for continuous monitoring on a societal and industrial scale produces vast amounts of data, enabling insights of unprecedented scale and granularity.

The work presented in this PhD reflects research into two subareas of computer vision: multi-modal analysis and robust traffic analysis. Within these areas, we have shed light on specific problems that we have analyzed in greater depth. As with many other aspects of life, taking a closer look reveals subtle details that cannot be seen with the naked eye.

1 Multi-Modal Analysis

Analysis of people from images or video is a vast field with numerous purposes. Automated analysis opens the door for large-scale surveillance and

behavioral studies used by authorities, companies, and researchers. In this thesis, we have conducted work within multi-modal people detection and re-identification which covers the identification of people from two or more cameras for which the views do not overlap. In airports and amusements parks, re-identification of people may be used to estimate queue lengths and how people move between facilities. In department stores and malls, the movement of people between shops is instrumental in maximizing the profits of the owners.

Traditionally, surveillance has relied on cameras that capture the visible light. Images from such cameras are easily interpretable for human inspectors but requires that the scene is sufficiently lit. In this thesis, we have explored the use of other imaging sensors to complement the traditional camera, such as thermal infrared and active stereo cameras. Thermal cameras capture the infrared radiation emitted from bodies. Two objects can be distinguished based on their difference in temperature and thus in the context of analysis of people, the temperature of the background must be significantly different from the persons of interest. Active stereo cameras, or depth cameras, actively emit light onto the scene and measure the distance from objects to the camera based on the reflected light. The promise of combining visible light, thermal, and depth cameras is increased redundancy and robustness to changes in lighting conditions, scene geometry, and temperature. Because the cameras rely on three disparate phenomena, challenges in one imaging domain may not necessarily affect the two other domains. An example of an indoor scene captured with visible light, thermal, and depth cameras is illustrated in Figure 1.1.

Besides looking at people from multi-modal sensors, we have also utilized multi-modal cameras in traffic surveillance, where the promise of continuous and robust monitoring of the road are challenged under bad weather and non-optimal lighting conditions.

2 Robust Traffic Analysis

The monitoring of road traffic serves many purposes. From a traffic security standpoint, observation of road users and their behavior is an essential tool for understanding the causation of accidents and eventually improve road safety [2, 5]. From a traffic management standpoint, understanding of road user behavior might be used to re-route traffic or optimize traffic flow.

Traditionally, monitoring of road traffic required that the traffic researcher was physically present at the observation site, manually taking notes of the traffic [4]. As one can imagine, this is a tedious task when conducted over several hours and the observation is a subjective, non-reversible process. The observer might miss some road users while watching others, and the

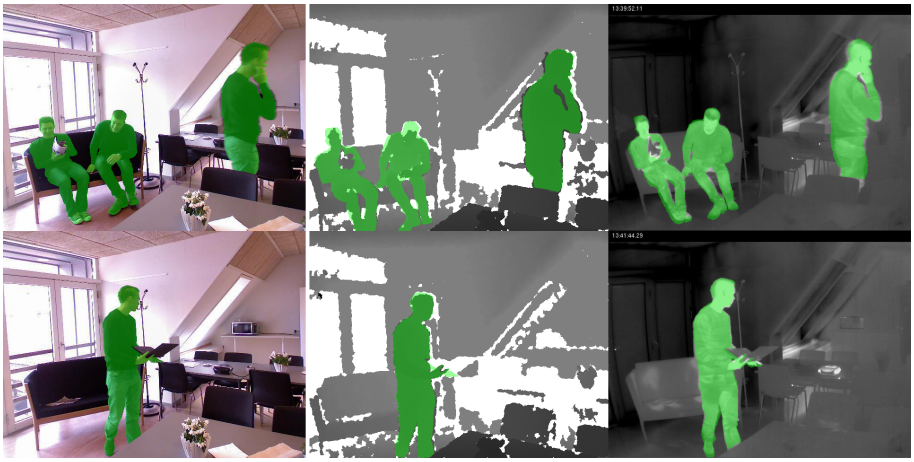


Fig. 1.1: Sample image from the AAU VAP Trimodal People Segmentation Dataset where an indoor scene is captured by a conventional camera, a depth camera, and a thermal camera.

preference on what to report or not might vary between observers.

Recording the traffic with a video camera allows the traffic researcher to study the details of interesting situations at her own pace and ensures the reproducibility of the research. However, the observation of hours and weeks of road user behavior remains a tiresome and tedious task. This is where automated traffic surveillance, enabled by computer vision, comes into the picture [1].

Automated traffic surveillance is a solved problem when the traffic is organized and there is a clear separation between road users, such as free-flowing traffic on highways. In urban traffic, however, complex intersections and multiple road user types leaves several challenges to computer vision algorithms. Furthermore, when the observability of the scene is impaired by low illumination and challenging weather conditions, it is increasingly hard for conventional vision algorithms to cope with the task of persistent traffic monitoring. An example of a traffic scene impaired by low illumination and reflections is shown in Figure 1.2.

A truly robust, automatic traffic monitoring system should be capable of solving the aforementioned challenges in a broad variety of road configurations and between varying road user types. There are several promising approaches to solving the problems: strengthening the core computer vision algorithms, pre-processing the raw images from the cameras, or using multiple, multi-modal sensors. A robust traffic monitoring system should possibly integrate elements from all three approaches.

Eventually, a traffic monitoring system must be usable for a general audience that does not necessarily has a background in computer vision. In this

(a) RGB **(b)** Thermal

Fig. 1.2: Traffic surveillance images from the AAU RainSnow dataset. The scene is captured by both a conventional visible light (RGB) camera and a thermal camera.

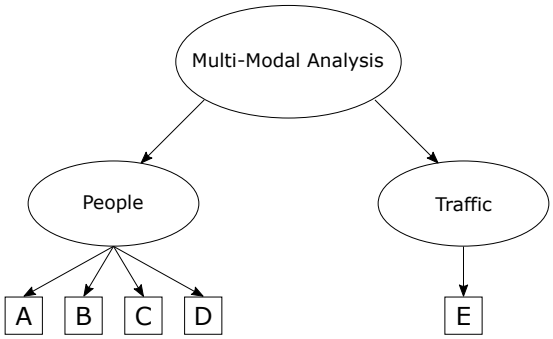


Fig. 1.3: Overview of the papers in this thesis covering multi-modal analysis. A box refers to a paper and a letter to the corresponding position in the appendix.

thesis, we describe this requirement as *robustness of use*.

3 Thesis Structure

In the following chapters, we will give an introduction to multi-modal analysis and robust traffic analysis, followed by a description of the state-of-the-art and our contributions to the respective fields.

This introduction will be followed by an appendix of three parts, each containing a collection of papers within a certain topic. Part II covers the work conducted in this thesis on Multi-Modal Analysis. As illustrated in Figure 1.3, this includes four papers on multi-modal analysis of people and one paper on multi-modal analysis of traffic.

Part III covers the work on Robust Traffic Analysis, containing three main

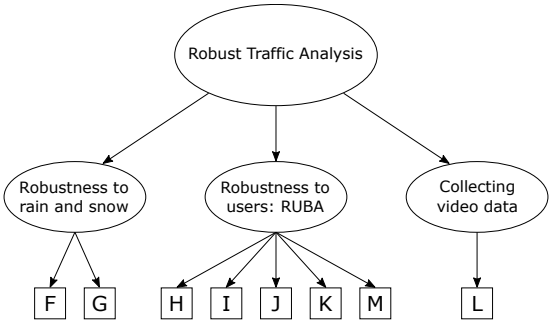


Fig. 1.4: Overview of the papers in this thesis covering robust traffic analysis. A box refers to a paper and a letter to the corresponding position in the appendix.

bodies of work:

- Robustness of computer vision algorithms to bad weather, more notably rain and snow.
- Robustness of use to traffic practitioners.
- Collecting video data using a portable system.

In Figure 1.4, these categories are linked to the respective papers.

Part IV covers the dissemination activities, more specifically an introduction to deep learning for the general public, published in the Danish magazine *Aktuel Naturvidenskab*.

References

[1] N. J. Ferrier, S. Rowe, and A. Blake, "Real-time traffic monitoring," in *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*. IEEE, 1994, pp. 81–88.

[2] C. Hydén, "The development of a method for traffic safety evaluation: The swedish traffic conflicts technique," *Bulletin Lund Institute of Technology, Department*, no. 70, 1987.

[3] S. A. Papert, "The summer vision project," 1966.

[4] M. Parker Jr and C. V. Zegeer, "Traffic conflict techniques for safety and operations: Observers manual," United States. Federal Highway Administration, Tech. Rep., 1989.

[5] E. Polders and T. Brijs, "How to analyse accident causation? a handbook with focus on vulnerable road users," *Deliverable 6.3. Horizon 2020 EC Project, InDeV*, 2018.

References

Chapter 2

Multi-Modal Analysis

1 Introduction

Traditionally, vision-based analysis has relied on conventional cameras to capture the world. These cameras capture the light visible to the human eye, i.e. light with wavelengths between 390 and 700 nm [51]. Seeing the world from one camera alone comes with some disadvantages, however. With only one view, it is impossible to estimate the absolute size of objects or the distance from objects to the camera.

1.1 Depth cameras

The addition of a second camera enables the computation of scale and depth through the underlying two-view geometry [19]. The computation of depth through two cameras is denoted *passive stereo vision* with the passive term stemming from the fact that the cameras are passively looking at the scene. Passive stereo relies on the texture of objects to compute disparity maps between the two views of the scene. The limitation of this approach is that the computation of depth is difficult with non-textured uniform surfaces and highly reflective objects.

Active stereo sensors circumvent these limitations by actively emitting light into the world and measuring the properties of the returned signal. However, the widespread usage of these sensors has been limited by high cost and low spatial resolution. This was changed by the introduction of the Microsoft Kinect sensor in 2010 [60]. The Kinect sensor combined a traditional visual camera with an emitter that projected near-infrared structured light onto a scene. An infrared camera observes the geometry of the projected pattern and calculates the depth on the basis of the observations. The Kinect sensor quickly became popular among researchers due to its low cost and ease-of-

use. The second generation of the Kinect, Kinect v2, replaced the infrared projector with a time-of-flight camera, enabling higher spatial resolution and accuracy [55].

Since the early days of the Kinect, active stereo sensors have been commoditized and integrated into several end-user applications. In user authentication, active stereo sensors are integrated into phones and computers to verify the user identify based on the three-dimensional face profile. In virtual and augmented reality, depth sensors such as the Microsoft Hololens and Intel PrimeSense [23] enable reliable and accurate tracking of users and objects.

1.2 Thermal cameras

Visual cameras require that the scene is sufficiently lit by an external light source. Whenever this requirement is violated, the cameras provide low-quality images and in total darkness, they will not be able to see anything. Thermal cameras, on the other hand, capture the infrared radiation of objects with a temperature above absolute zero and may thus operate in total darkness and non-optimal illumination conditions. In the scope of this thesis, we will use thermal cameras that capture radiation in the long-wavelength infrared (LWIR) spectrum, covering wavelengths from 8 to 15 μm [16]. As opposed to the visible spectrum, infrared radiation does not travel through glass. Objects with the same temperature as the background will be largely invisible on the thermal image whereas they might be immediately recognized from a visual image by their texture and shape.

Thermal cameras are a compelling choice for the observation of people. Humans and animals are easily recognizable if their temperature is significantly different from the background, which most often is the case in temperate and sub-tropical climates. Because objects in thermal images carry fewer discriminative features than visual images, it is harder to identify people based on their thermal signature. With increasing concerns for privacy, thermal cameras are a favorable choice.

1.3 Combining the sensors

As described in the above, visual, thermal, and depth cameras sense the world using disparate sensors, each sensitive to different conditions. Because of the disparate nature of the sensors, a combination of them would increase the robustness of the system. A detrimental condition in one modality, such as bad lighting in the visual domain, will not impair the thermal and depth domains whereas inanimate objects that are hard to recognize in the thermal domain might be easier to detect in the visual and depth domains.

Before one can harvest the benefits of multi-modal sensors, at least two issues must be dealt with:

- The multi-modal images must be registered, i.e. one should know how to transfer a spatial location from one modality to another. The levels of granularity might vary depending on the application: it might be enough to roughly align the position of an object based on its contours or it might be necessary to perform a pixel-wise registration of any position in the scene.
- Once registered, the information of the modalities should be combined, either by low-level fusion of the image streams, mid-level fusion at algorithmic level, or by late decision-level fusion.

2 State-of-the-art

In the following, we will provide an overview of recent research within multi-modal analysis. We will limit ourselves to works that relate to the issues mentioned in the section above, i.e. registration and information fusion of multi-modal images. However, any research on the mentioned issues rely on the acquisition of multi-modal data. The widespread adoption of the Kinect and related hardware has led to an increase in the availability of visual and depth datasets. Registration of visual-depth imagery is eased by the fact that the visual and depth cameras are built into the same hardware, and the sensors are typically calibrated from the factory. With the availability of accurate calibration, the additional depth data makes it possible to transfer points from two-dimensional image coordinates in one domain to three dimensional world coordinates and back to two-dimensional image coordinates in the second domain.

In datasets without any depth information, such as visual-thermal datasets, it is not possible to transfer a position from one domain to another without any further assumptions. In the following subsection, we will describe the most commonly used assumptions in visual-thermal registration.

In Table 2.1, we have listed the publicly available datasets containing images and videos from both the visual and thermal domains. An overview of available datasets containing visual and depth information is available in [34].

Table 2.1: Multi-modal datasets in computer vision, containing images from the thermal and visual domains. Adapted from [11, 32, 59].

Dataset	Year	Sensor placement	Location	Purpose	Visual	Depth	Thermal	Number of images per modality
AAU RainSnow [4]	2018	Top view	Outdoor	Road user detection	X		X	≈ 130,800
KAIST2018 [11]	2018	Vehicle	Outdoor	Pedestrian detection	X	X	X	8,970
VLIRVDIF [14]	2017	Head height	Mixed	People detection	X		X	≈ 56,820
Color and Thermal Stereo Dataset (CATS) [53]	2016	Head height	Outdoor	People detection	X	X	X	1,372
VAP Trimodal People Segmentation Dataset [42]	2016	Head height	Indoor	People segmentation	X	X	X	5,924
CVC-14 [17]	2016	Vehicle	Outdoor	Pedestrian detection	X		X	8,473
GTD [28]	2016	Mixed	Outdoor	Surveillance	X		X	7,850
KAIST2015 [20]	2015	Vehicle	Outdoor	Pedestrian detection	X		X	50,184
Bilodeau [9]	2014	Head height	Indoor	People detection	X		X	7,821
CVC-15 [5]	2013	Vehicle	Outdoor	Calibration	X	X	X	100
LITIV2012 [52]	2012	Top view	Indoor	People tracking	X		X	6,325
INO Video Analytics Dataset [1]	2012	Top view	Outdoor	Surveillance	X		X	12,680
OSU Color-Thermal Database [13]	2007	Top view	Outdoor	Pedestrian detection	X		X	8,545
Bristol Eden Multi-Sensor Data Set [27]	2006	Top view	Outdoor	People detection	X		X	1,700

It is apparent from Table 2.1 that the availability of visual-thermal image pairs is limited compared with the abundance of databases containing visual images. The lack of data is even more noticeable when looking for annotated imagery in the visual-thermal domain. Because such datasets are very sparse compared to the millions of annotated visual images in ImageNet [47] and COCO [30], researchers have started to investigate how to transfer annotated visual imagery into the thermal domain and vice versa.

In [8], Berg *et al.* used a convolutional neural network (CNN) with a U-Net structure to generate visual images from input images in the thermal domain. Inspired by the recent success of generative adversarial networks for image-to-image translation [21], Zhang *et al.* [59] trained a generative network on available, not necessarily annotated, visual-thermal image pairs. The generative network is used to synthesize the corresponding thermal images from annotated images in the visual domain. The extended, synthetic dataset is used to train a deep convolutional neural network from scratch. Their results show that training the network on the synthetic and real thermal images gives a significant increase in performance when compared with training on the limited amount of real thermal images.

2.1 Registration

The most commonly used technique for registering the thermal and visible domains is by using a single homography, a two-dimensional image transformation applied on images in one domain to spatially align them into the second domain. It is easy to compute the homography as it only requires four corresponding point pairs. However, its usage relies on the assumption that objects of interest in a scene are lying on the same virtual plane. In many surveillance scenarios, this is often the case when the cameras are observing the scene from a considerable height and the objects of interest are moving on the ground.

However, the assumption breaks down if the distance from the camera to the virtual plane is not at least an order of magnitude larger than the distance from the objects to the virtual plane. Notable use cases that violates the planar assumption are vehicle-mounted sensors and indoor surveillance scenarios. In such situations, one must use other clues to search for correspondence. Krotosky and Trivedi [25] provide an excellent overview of the available registration and alignment techniques, and we refer the reader to their work for an overview of the geometrical concepts of multi-modal image registration.

The same authors provided in [26] a dual stereo setup containing two visual and two thermal sensors for pedestrian detection in vehicles. The authors calibrated the setup using a heated calibration board and used the trifocal tensor to register pixels from one modality to another. Detection of pedestrians was performed via Histograms of Oriented Gradients (HOG) [12]

and classified using a Support Vector Machine (SVM).

In [49], Shibata *et al.* propose a novel calibration target for visual-thermal calibration that consists of a checkerboard where the white squares are raised from the black backdrop.

Torabi *et al.* [52] align the thermal and visual images using an affine transformation and fuse information from the two modalities using a silhouette matching approach. In subsequent work from the same research group, St. Charles *et al.* [50] use the epipolar constraint to restrict the search for corresponding points in the visible and thermal modalities. A sliding window approach is used to restrict the correspondence search and a match is found based on the appearance and shape of the windows. Evaluation is performed on the VAP Trimodal People Segmentation [42] and Bilodeau [9] datasets. The registered data is used to segment the foreground based on color, contour, smoothness, and temporal properties.

Chen *et al.* [10] use Speeded-Up Robust Features (SURF) [7] descriptors on both modalities to register the detected points by using graph matching. However, this only holds when the visual and thermal signatures of the objects of interest are largely similar.

Registration might be avoided if the light enters the sensors at the same location. This is possible by using a beam-splitter to reflect the incoming light into the visual and thermal sensors. Such a setup is used for the acquisition of the KAIST datasets [20] which produces an almost perfect spatial alignment between the visual and thermal images. However, Berg *et al.* [8] discovered in a recent study that the alignment was slightly inaccurate, resulting in displacement errors of up to four pixels in the vertical direction and up to 16 pixels in the horizontal direction.

2.2 Data fusion

When the registration is accomplished and the images from the sensors are both spatially and temporally aligned, one should decide how to combine the information from the two sensors.

The authors behind the CVC-15 dataset [5] use a combination of HOG and Local Binary Patterns (LBP) [40], trained using either a SVM classifier, a Random Forest classifier, or a Deformable Part Model [15]. Registration of the two modalities is performed on a bounding-box level and descriptors are computed using the combined imagery. The authors found that the fused detector improved the overall results on the daytime images whereas the detector trained solely on thermal images outperformed the corresponding fused and visual detectors at nighttime.

In [46], the authors experiment with feature descriptors on the thermal and visual domains and find that the Scale-Invariant Feature Transform (SIFT) [31] descriptor provide good results in the thermal domain. Image fusion is

performed in [54] by the use of a Guided filter under the assumption that the images are already aligned. The authors of [6] use a similar assumption and fuse the modalities using a saliency rule on filtered images.

Li *et al.* [29] experiment with six fusion schemes on the KAIST2015 data [20] by using a Faster R-CNN model [45]. The fusion schemes range from input fusion to late fusion after the FC7 layer [24] and the experimental results suggest that mid-level and late FC7-level fusion provide the best overall results. However, at nighttime with low illumination, the authors found that the stand-alone thermal modality outperformed all fusion schemes. This is similar to the findings from [5] and suggests that the information from visual images should be largely disregarded under bad illumination conditions.

Multi-Modal Re-Identification

Until recently, the research community has shown little interest in using thermal cameras for person re-identification. However, two recent datasets containing each 491 and 412 persons has sparked interest within the field [38, 57]. In the former dataset, Wu *et al.* [57] perform experiments on how to effectively fuse the visual and thermal images using a CNN. They propose an architecture where the visual image is converted to a one-channel grayscale image. The thermal image is added as the second channel whereas the third channel is filled with zeros. This setup permits the use of CNNs that are pre-trained on three-channel color images. In the latter dataset, Nguyen *et al.* [38] extract CNN and HOG features separately for each modality and concatenate them in a late fusion step. Ye *et al.* [58] build upon the two datasets and perform late fusion of thermal and visual imagery by training two separate CNNs and fusing the two streams after the FC7 layer.

The use of depth information is more widespread within person re-identification. One traditional approach to describing the information in a depth image is to convert popular feature representations in the visual domain to the depth domain. For instance, HOG features are used in [33] to match people in a depth-only re-identification setup. Another approach is to describe the depth information as point clouds. The authors behind [56] propose new shape descriptors in the depth domain based on the co-variance of depth voxes. The depth features are subsequently fused with conventional HOG and LBP features in the visual domain to perform re-identification. One may also infer the position of the body joints from the depth signature. The joints can be used for calculating anthropometric measures which is used in [41] to perform re-identification.

More recently, the use of CNNs has also gained traction within depth-based person re-identification. In the work of Ren *et al.* [44], two CNNs are used to extract features separately from visual and depth images. The latent variables of the two CNNs, comparable to the FC7 features used in [58], are

subsequently concatenated into a joint feature vector. However, when the visibility of one modality is impaired, a fused system will lose information in half of the feature vector, severely affecting the performance. A way to mitigate this is to create a shared embedding between the depth and visual modalities which means that a person should ultimately be represented by the same feature vector, regardless of whether the person is imaged by a visual or a depth camera. Such cross-modal feature embeddings have been proposed in [18] to re-identify persons from depth images even though they have only been observed previously by a visual camera.

3 Contributions

Our interest in multi-modal imagery stems from a desire to combine the ongoing work within thermal imaging with conventional visual cameras and recent advantages in depth sensing technologies, most notably the Kinect. We initially estimated that it would take a few months to acquire synchronized data from the three modalities and register the data accordingly. It eventually turned out much more difficult to acquire a synchronized dataset from such disparate sensors, leaving many iterations of trial-and-error before the final datasets were successfully captured. The synchronization process was solved by saving time stamps for each frame of every modality and matching the frames in a post-processing step.

The calibration and registration of the sensors were an integral part of the acquisition process due to the following observations:

1. Conventional calibration checkerboards may not be used as-is due to low contrast in the thermal domain.

2. Because the objects were near the imaging sensors, a planar homography would fail to accurately register the modalities. Thus, we had to rely on a dense cluster of point-pairs in the three modalities to estimate the calibration and registration parameters.

The acquisition process is described in more detail in my master's thesis [3]. The acquisition platform was used to capture a dataset for tri-modal people re-identification published in [35]. To the best of our knowledge, the work was the first to integrate visual, thermal, and depth imagery in re-identification. The corresponding paper is included in Appendix A. Our acquisition platform was also used to record a dataset for multi-modal face recognition [39].

Our efforts on tri-modal image acquisition and registration resulted in a joint work with Universitat de Barcelona on segmentation of people [42]. The work was based on the AAU VAP Trimodal People Segmentation Dataset which consisted of three indoor scenes captured with visible, depth, and

thermal cameras. The dataset is publicly available on Kaggle[1] and is, to the best of our knowledge, one of the first publicly available tri-modal datasets containing people. See Appendix B for further details.

Because the dataset should be used for people detection and segmentation, is was a necessity to annotate each individual image frame on a pixel level. A former PhD student in our lab, Andreas Møgelmose, built an annotation toolbox specifically designed for annotating the tri-modal imagery. I adopted the toolboxes for pixel-wise annotation and bounding box annotation from Andreas and made them available as open source tools to the general public. Since then, I have extended the annotation toolboxes to handle a large range of different use cases and datasets, ranging from single-modal to multi-modal applications. Within our laboratory, the toolboxes have been used for annotating chicken entrails [43], road users [2,4], pigs, material defects, fish [22], and basketball players [48]. The annotation toolboxes are described in more detail in Appendix D.

The work within person re-identification has been further developed in the visual and depth domains [36]. The work is an extension of [37] with two new datasets and a more robust method for handling the depth data from the Kinect. Find more details in Appendix C.

I was also co-supervising a master's student who was investigating how to combine visual and thermal imagery for traffic surveillance [2]. We investigated the use of several metrics for judging the quality of the visual-thermal images, both internal metrics such as the image entropy but also contextual metrics such as the position of the sun and the current weather condition. Our joint article is included in Appendix E.

An overview of the work conducted within multi-modal analysis is given in Figure 2.1. The illustration shows the relationships between the published articles, datasets, and programs developed during my PhD. A similar illustration for the work within robust traffic analysis is found in Figure 3.3.

4 Sub-conclusion

Our main contributions within multi modal analysis have been the following:

- Our work within acquisition, registration, and synchronization of visual, depth, and thermal video which has been applied in three papers of this thesis.
- The publicly available AAU VAP Trimodal People Segmentation Dataset consisting of three indoor scenes captured with visible, depth, and thermal cameras.

[1] `https://www.kaggle.com/aalborguniversity/trimodal-people-segmentation`

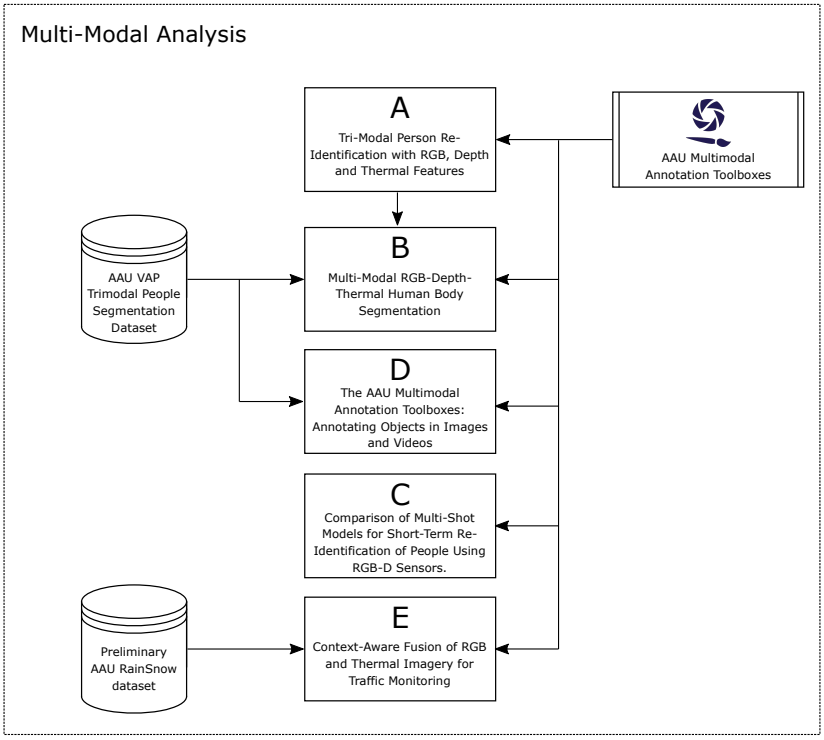


Fig. 2.1: The structure of the work within multi-modal analysis of this thesis. Boxes represents papers and reports. The letters within the boxes refer to the corresponding position in the appendix of this thesis.

- The work on estimating quality metrics for combining visual and thermal images for traffic surveillance.
- The extension and development of the annotation toolboxes which have been useful for other researchers in our laboratory.

References

[1] "Ino video analytics dataset," https://www.ino.ca/en/examples/video-analytics-dataset/, 2012.

[2] T. Alldieck, C. H. Bahnsen, and T. B. Moeslund, "Context-aware fusion of rgb and thermal imagery for traffic monitoring," *Sensors*, vol. 16, no. 11, p. 1947, 2016.

[3] C. Bahnsen, "Thermal-visible-depth image registration," *Unpublished Master Thesis, Aalborg University, Aalborg, Denmark*, 2013.

[4] C. H. Bahnsen and T. B. Moeslund, "Rain removal in traffic surveillance: Does it matter?" *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–18, 2018.

[5] F. Barrera, F. Lumbreras, and A. D. Sappa, "Multispectral piecewise planar stereo using manhattan-world assumption," *Pattern Recognition Letters*, vol. 34, no. 1, pp. 52–61, 2013.

[6] D. P. Bavirisetti and R. Dhuli, "Two-scale image fusion of visible and infrared images using saliency detection," *Infrared Physics & Technology*, vol. 76, pp. 52–64, 2016.

[7] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[8] A. Berg, J. Ahlberg, and M. Felsberg, "Generating visible spectrum images from thermal infrared," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018, pp. 1143–1152.

[9] G.-A. Bilodeau, A. Torabi, P.-L. St-Charles, and D. Riahi, "Thermal–visible registration of human silhouettes: A similarity measure performance evaluation," *Infrared Physics & Technology*, vol. 64, pp. 79–86, 2014.

[10] Y. Chen, X. Zhang, F. Li, and Y. Zhang, "Multi-modal image registration based on modified-surf and consensus inliers recovery," in *International Conference on Image and Graphics*. Springer, 2017, pp. 612–622.

[11] Y. Choi, N. Kim, S. Hwang, K. Park, J. S. Yoon, K. An, and I. S. Kweon, "Kaist multi-spectral day/night data set for autonomous and assisted driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 934–948, 2018.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[13] J. W. Davis and V. Sharma, "Background-subtraction using contour-based fusion of thermal and visible imagery," *Computer vision and image understanding*, vol. 106, no. 2-3, pp. 162–182, 2007.

[14] A. Ellmauthaler, C. L. Pagliari, E. A. da Silva, J. N. Gois, and S. R. Neves, "A visible-light and infrared video database for performance evaluation of video/image fusion methods," *Multidimensional Systems and Signal Processing*, pp. 1–25, 2017.

[15] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[16] R. Gade and T. B. Moeslund, "Thermal cameras and applications: a survey," *Machine vision and applications*, vol. 25, no. 1, pp. 245–262, 2014.

[17] A. González, Z. Fang, Y. Socarras, J. Serrat, D. Vázquez, J. Xu, and A. M. López, "Pedestrian detection at day/night time with visible and fir cameras: A comparison," *Sensors*, vol. 16, no. 6, p. 820, 2016.

[18] F. Hafner, A. Bhuiyan, J. F. Kooij, and E. Granger, "A cross-modal distillation network for person re-identification in rgb-depth," *arXiv preprint arXiv:1810.11641*, 2018.

[19] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[20] S. Hwang, J. Park, N. Kim, Y. Choi, and I. So Kweon, "Multispectral pedestrian detection: Benchmark dataset and baseline," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1037–1045.

[21] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 5967–5976.

[22] A. Karpova and J. B. Haurum, "Re-identification of zebrafish using metric learning," *Unpublished Master Thesis, Aalborg University, Aalborg, Denmark*, 2018.

[23] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, "Intel realsense stereoscopic depth cameras," *arXiv preprint arXiv:1705.05548*, 2017.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[25] S. J. Krotosky and M. M. Trivedi, "Mutual information based registration of multimodal stereo videos for person tracking," *Computer Vision and Image Understanding*, vol. 106, no. 2-3, pp. 270–287, 2007.

[26] ——, "On color-, infrared-, and multimodal-stereo approaches to pedestrian detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 4, pp. 619–629, 2007.

[27] J. Lewis, S. Nikolov, A. Loza, E. F. Canga, N. Cvejic, J. Li, A. Cardinali, C. Canagarajah, D. Bull, T. Riley *et al.*, "The eden project multi-sensor data set," *The Online Resource for Research in Image Fusion (ImageFusion. org)*, 2006.

[28] C. Li, H. Cheng, S. Hu, X. Liu, J. Tang, and L. Lin, "Learning collaborative sparse representation for grayscale-thermal tracking," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5743–5756, 2016.

[29] C. Li, D. Song, R. Tong, and M. Tang, "Illumination-aware faster r-cnn for robust multispectral pedestrian detection," *Pattern Recognition*, vol. 85, pp. 161–171, 2019.

[30] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[31] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[32] C. Luo, B. Sun, Q. Deng, Z. Wang, and D. Wang, "Comparison of different level fusion schemes for infrared-visible object tracking: An experimental survey," in *2018 2nd International Conference on Robotics and Automation Sciences (ICRAS)*. IEEE, 2018, pp. 1–5.

[33] M. Madadi, S. Escalera, J. Gonzalez, F. X. Roca, and F. Lumbreras, "Multi-part body segmentation based on depth maps for soft biometry analysis," *Pattern Recognition Letters*, vol. 56, pp. 14–21, 2015.

[34] L. Maddalena and A. Petrosino, "Background subtraction for moving object detection in rgbd data: A survey," *Journal of Imaging*, vol. 4, no. 5, p. 71, 2018.

[35] A. Mogelmose, C. Bahnsen, T. Moeslund, A. Clapes, and S. Escalera, "Tri-modal person re-identification with rgb, depth and thermal features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 301–307.

[36] A. Møgelmose, C. Bahnsen, and T. B. Moeslund, "Comparison of multi-shot models for short-term re-identification of people using rgb-d sensors," in *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications 2015Computer Vision Theory and Applications*. SCITEPRESS Digital Library, 2015, pp. 244–251.

[37] A. Møgelmose, T. B. Moeslund, and K. Nasrollahi, "Multimodal person reidentification using rgb-d sensors and a transient identification database." in *IWBF*, 2013, pp. 1–4.

[38] D. T. Nguyen, H. G. Hong, K. W. Kim, and K. R. Park, "Person recognition system based on a combination of body images from visible light and thermal cameras," *Sensors*, vol. 17, no. 3, p. 605, 2017.

[39] O. Nikisins, K. Nasrollahi, M. Greitans, and T. B. Moeslund, "Rgb-dt based face recognition," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 1716–1721.

[40] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[41] F. Pala, R. Satta, G. Fumera, and F. Roli, "Multimodal person reidentification using rgb-d cameras," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 4, pp. 788–799, 2016.

[42] C. Palmero, A. Clapés, C. Bahnsen, A. Møgelmose, T. B. Moeslund, and S. Escalera, "Multi-modal rgb–depth–thermal human body segmentation," *International Journal of Computer Vision*, vol. 118, no. 2, pp. 217–239, 2016.

[43] M. P. Philipsen, J. V. Dueholm, A. Jørgensen, S. Escalera, and T. B. Moeslund, "Organ segmentation in poultry viscera using rgb-d," *Sensors*, vol. 18, no. 1, p. 117, 2018.

[44] L. Ren, J. Lu, J. Feng, and J. Zhou, "Multi-modal uniform deep learning for rgb-d person re-identification," *Pattern Recognition*, vol. 72, pp. 446–457, 2017.

[45] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[46] P. Ricaurte, C. Chilán, C. A. Aguilera-Carrasco, B. X. Vintimilla, and A. D. Sappa, "Feature point descriptors: Infrared and visible spectra," *Sensors*, vol. 14, no. 2, pp. 3690–3701, 2014.

[47] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[48] A. A. Sangüesa, T. B. Moeslund, C. H. Bahnsen, and R. B. Iglesias, "Identifying basketball plays from sensor data; towards a low-cost automatic extraction of advanced statistics," in *Data Mining Workshops (ICDMW), 2017 IEEE International Conference on*. IEEE, 2017, pp. 894–901.

[49] T. Shibata, M. Tanaka, and M. Okutomi, "Accurate joint geometric camera calibration of visible and far-infrared cameras," *Electronic Imaging*, vol. 2017, no. 11, pp. 7–13, 2017.

[50] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "Online mutual foreground segmentation for multispectral stereo videos," *arXiv preprint arXiv:1809.02851*, 2018.

[51] C. Starr, C. Evers, and L. Starr, *Biology: concepts and applications without physiology*. Cengage Learning, 2010.

[52] A. Torabi, G. Massé, and G.-A. Bilodeau, "An iterative integrated framework for thermal–visible image registration, sensor fusion, and people tracking for video surveillance applications," *Computer Vision and Image Understanding*, vol. 116, no. 2, pp. 210–221, 2012.

[53] W. Treible, P. Saponaro, S. Sorensen, A. Kolagunda, M. O'Neal, B. Phelan, K. Sherbondy, and C. Kambhamettu, "Cats: A color and thermal stereo benchmark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2961–2969.

[54] X. Wang, R. Nie, and X. Guo, "Two-scale image fusion of visible and infrared images using guided filter," in *Proceedings of the 7th International Conference on Informatics, Environment, Energy and Applications*. ACM, 2018, pp. 217–221.

[55] O. Wasenmüller and D. Stricker, "Comparison of kinect v1 and v2 depth images in terms of accuracy and precision," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 34–45.

[56] A. Wu, W.-S. Zheng, and J.-H. Lai, "Robust depth-based person re-identification." *IEEE Trans. Image Processing*, vol. 26, no. 6, pp. 2588–2603, 2017.

[57] A. Wu, W.-S. Zheng, H.-X. Yu, S. Gong, and J. Lai, "Rgb-infrared cross-modality person re-identification," 2017.

[58] M. Ye, Z. Wang, X. Lan, and P. C. Yuen, "Visible thermal person re-identification via dual-constrained top-ranking." in *IJCAI*, 2018, pp. 1092–1099.

[59] L. Zhang, A. Gonzalez-Garcia, J. van de Weijer, M. Danelljan, and F. S. Khan, "Synthetic data generation for end-to-end thermal infrared tracking," *arXiv preprint arXiv:1806.01013*, 2018.

[60] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE multimedia*, vol. 19, no. 2, pp. 4–10, 2012.

Chapter 3

Robust Traffic Analysis

1 Introduction

In 2015, 26.000 people were killed in road accidents in the European Union, accounting for 0.5 % of all deaths in the member states [20]. In order to reduce these numbers, it is instrumental to understand the causation of the accidents. Observing the road through video analysis is one way of gaining insight into the behavior of road users and their interactions. The insights may ultimately lead to improvements in infrastructure layout and vehicle design and could suggest changes in legislation and policies.

In the following, we will describe the challenges concerning observation of the road and analysis of the road users. We focus on infrastructure-side monitoring of the road, i.e. observing the road with a stationary camera that might be placed on a mast, building, or similar existing infrastructure. The development of autonomous vehicles has sparked a huge interest in vehicle-side monitoring of the road. The two fields share many of the same methods and techniques and may share information in the future via vehicle-to-infrastructure communication.

1.1 Observing the road

There are several issues to consider when designing a system for road traffic observation. The layout of the road to observe, the types of road users to detect, and the reliability requirements of the overall system. A robust system for observing the road should feature a high reliability and resilience to phenomena such as occlusion and the effects of varying illumination and weather conditions. One of the focal points of this thesis is to study the effects of the weather and how detrimental effects might be mitigated. Sample observations under rainfall and snowfall are shown in Figure 3.1.

(a) Snow **(b)** Heavy rain

(c) Raindrops on the lens **(d)** Reflections

Fig. 3.1: Examples of different weather phenomena in traffic surveillance as observed by visual and thermal cameras. Images are from the AAU RainSnow dataset [15].

In Table 3.1, we list the effects of weather on surveillance conducted with either a visual or a thermal long-wavelength infrared (LWIR) camera. In general, the concentration of water in the air leads to reduced visibility in both the visual and thermal domains, regardless of whether the water comes as precipitation such as snow or rain or by the quasi-static appearance of fog and haze. In general, both domains are affected by the scattering and absorption of light from particles, with subtle differences in how the modalities are affected by the specific phenomena. Light haze has almost no impact on the visibility from thermal cameras whereas the attenuation from rain is the same in both the visible and LWIR domains [17]. The attenuation due to falling snow in the visual domain is 5 to 20 times larger than raindrops for the same mass precipitation rate [43]. However, the attenuation due to falling snow is $30 - 50$ % higher in the LWIR than the visual domain, due to the relatively large snow crystals and larger wavelengths in the LWIR domain [43].

1.2 Analyzing the traffic

Once a reliable observation of the road has been established, one should decide how to use the observations. If the goal is to improve the safety of the road, it makes sense to analyze the situations in which accidents occur. However, because accidents are sparse events that happen very infrequently, it is unfeasible to wait several years for the accidents to happen. Should an accident happen, the situation might even occur at a position not covered by a

Table 3.1: Weather effects on traffic observation with visual and thermal sensors. Adapted from [15] and based on [17, 43, 48].

Phenomena	Effect in the visual domain	Effect in the thermal domain
Raindrops on lens	Blur, diffuse scattering of light, especially from headlights	Blur, reduced visibility, not susceptible to scattering from headlights
Dense rain	Spatio-temporal streaks affecting visibility, reduction in visible range	Streaks not visible, reduction in visible range
Dense snow	Spatio-temporal streaks affecting visibility, reduction in visible range	Streaks not visible, severe reduction in visible range
Water on the road	Visible signature reflected	Heat signature reflected
Shadows	Differences in illumination	Persistent shadows visible due to differences in temperature
Light haze	Reduced visibility	Almost no effect
Light fog	Reduced visibility	Slightly reduced visibility
Heavy fog	Severely reduced visibility	Severely reduced visibility

surveillance camera.

In order to understand the causation of accidents without waiting for them to happen, traffic researchers study the dangerous situations in the traffic, known as conflicts. Conflicts are defined as events that would have resulted in an accident if at least one of the road users did not perform an evasive action [25]. Researchers from Lund University has shown that the severity and volume of these conflicts are correlated with the number and severity of accidents [25, 46] and such conflicts can therefore be used as surrogate safety indicators.

Besides safety and behavioral analysis, there are numerous other applications for traffic analysis such as automated tolling [3], speed enforcement [2], congestion detection [13], congestion prediction [47], and occupancy analysis of parking spaces [10].

2 State-of-the-art

In the following section, we will give an overview of the state-of-the-art within robust observation of the road, covering computer vision and signal processing approaches for improving the observability of the road. Hereafter, we will give an introduction to robust methods for traffic analysis, covering tools and solutions that can be useful for practitioners within traffic research and traffic management.

An overview of methods for data collection is available in Appendix L.

2.1 Robust observation of the road

As described in the introduction in Chapter 1, there are several approaches to solving the problem of robust traffic analysis:

- Pre-processing the input images and video to improve the signal-to-noise ratio.
- Strengthening the core vision algorithms in an integrated manner.
- Obtaining robustness by redundancy, for instance by using multiple, multi-modal sensors.

We have given an introduction to the latter in Chapter 2 on multi-modal analysis. The two remaining approaches are described below.

Pre-processing the input by de-weathering

In general, pre-processing algorithms aim to remove unwanted noise and artefacts in images and videos such that:

1. The output image is more pleasing to a human observer.
2. The signal-to-noise ratio is increased such that the performance of subsequent computer vision algorithms improve.

In traffic surveillance, one might want to mitigate the detrimental effects of bad weather. We denote such pre-processing algorithms as de-weathering algorithms. If we look at the phenomena described in Table 3.1, we can define two sub-fields within de-weathering algorithms:

- **Dehazing and defogging**
  Haze, fog, and mist are quasi-static phenomena that remain largely unchanged for seconds and minutes. The visibility of a scene is degraded by the accumulation of water and other particles in the air that scatter and attenuate the incident light. Far-away objects are more affected by haze and fog than objects close to the camera. An effective method for the removal of haze and fog might thus be guided by the depth of a scene. We refer to the reviews by Li *et al.* [34] and Lee *et al.* [32] for an overview of dehazing and defogging algorithms.
- **Rain and snow removal**
  Rain and snow streaks are visible as spatio-temporal fluctuations that temporarily occlude parts of the scene. The accumulation of dense rain and snow may also lead to degradation of the overall visibility of a scene, comparable to the effects of fog and haze. We review the techniques for rain and snow removal in [15] which is included in Appendix F.

In addition to the mitigation of the atmospheric conditions listed above, there exists many other approaches for removing unwanted artefacts. Some examples are the detection and removal of shadows [42] and reflections [21] as well as the detection of headlights from oncoming vehicles [33].

If the aforementioned pre-processing methods shall be effective, one should know when and where to apply them. For instance, algorithms for detection and removal of shadows might be useless under overcast weather conditions where there are no shadows. Thus, one would need to assess the current state of the weather, either by analyzing the image [14, 24, 35] or by incorporating information from external sources [11, 29]. Furthermore, it is useful to assess the influence of the weather on computer vision methods. In the work by Duthon *et al.* [19], the performance of eight commonly used image features are compared under different levels of natural and simulated rain. The results showed that some features, such as edge-based features and CNN-based features [44], breaks down under the presence of heavy rainfall.

Strengthening the core vision algorithms

An excellent overview of computer vision based methods for urban traffic is found in the work of Buch *et al.* [16]. A more recent overview of vision-based analysis at traffic in intersections is given by Datondji *et al.* [18] who list approaches for the detection and tracking of objects. The authors categorize object detection methods into four sub-domains:

- *Background subtraction* which contains approaches that maintains a model of the quasi-static scene background and the moving foreground objects.
- *Feature-based segmentation*, building on spatial features to obtain an object model. Datondji *et al.* only list approaches that use hand-crafted features, but more recent methods using CNNs fit into this category as well.
- *Model-based segmentation* which uses the calibrated three-dimensional geometry of the scene to search for objects that fit into pre-defined volumes.
- *Motion-based segmentation*, based on the optical flow between consecutive frames to distinguish moving objects from the background.

All of the mentioned approaches for object detection may be coupled with a tracking scheme for consistent spatio-temporal trajectories. A thorough overview of the state-of-the-art within the above-mentioned sub-domains and the promising directions to strengthen the methods are considered to be out of scope of this thesis. However, we notice that many advancements in computer vision within the last decade is based on the availability of large-scale public datasets and competitions. A list of datasets for urban traffic

surveillance is found in both surveys [16, 18]. Recent datasets not included in the two surveys are ChangeDetection.net [49], UA-DETRAC [39], MIO-TCD [38], and the NVIDIA AI City Challenge [40]. Efforts to strengthen the algorithms should be validated through standardized testing through such datasets, which hopefully will contain an increasingly large subspace of real world variation.

2.2 Robust methods for traffic analysis

In this subsection, we will focus on systems for automated traffic surveillance which provide useful metrics for practitioners within traffic management and traffic safety analysis. The focal point of this overview is on the interdisciplinary approaches that couple expertise within computer vision with the field of applied traffic analysis.

As mentioned in the introduction, a good framework for traffic analysis should feature both a robust system for detection, classification, and tracking of road users as well as being accessible for practitioners without a background in computer vision. Below, we will give a non-exhaustive overview of existing frameworks for automated analysis of road traffic. We will list both academic and commercial frameworks:

- *TrafficIntelligence* [26] is a collection of command-line tools for detection, classification, and tracking of road users. Objects are detected using Lukas-Kanade features [37] that are grouped based on distance criteria. The software is maintained by Polytechnique Montréal and contains a vast number of tools for detailed behavioral studies.

- *Urban Tracker* [28] is a joint-venture between the authors of TrafficIntelligence and the LITIV laboratory at Polytechnique Montréal. The system consists of a graphical user interface that allows the user to configure the parameters for object detection and tracking. Background subtraction is used to maintain a model of the moving objects in the scene which are subsequently grouped and tracked. The system is unable to classify objects and the reliance on background subtraction means that some objects are either detected as several road users or that two or more adjacent road users are detected as one. The purpose of Urban Tracker is to function as a watch-dog, guiding an practitioner to situations of interest.

- *RUBA* [9] is an academic tool for practitioners that serves as a watch-dog for situations of interest. Based on relatively simple computer vision tools, the system features four detectors that register either object presence, object motion, stationary objects, and states of a traffic light. Practitioners might combine these detectors to look out for interesting

situations such as red-running vehicles or potential conflicts between road users.

- *T-analyst* from Lund University [8] is a software tool for the manual creation of road user trajectories. For obtaining reliable metrics, the surface of the road is registered to real-world metric coordinates. Then, the user may adjust wire-frame boxes to fit the road user for every n frames. The obtained trajectories may be used for surrogate safety analysis.

- *STRUDL* [27] is an emerging framework from Lund University that uses CNNs for object detection and an Hungarian algorithm for tracking of the detected objects. The tool is currently command-line based and relies on GPUs to obtain real-time performance.

- *Pedtrax & SmartCycle* are two commercial products from Iteris [1] that detect and count pedestrians and bicyclists at intersections. The products are integrated in the camera surveillance services of the company, offering a plug-and-play solution.

- *Traffic Analytics* from Microsoft [12] is a cloud-based solution for providing directional counts of road users at intersections. Background subtraction is used to guide a CNN-based object detector that classifies road users into vehicles, pedestrians, and cyclists which are tracked by a unspecified method. The system provides a web-based interface for visualizing the results.

- *OTUS3D* from Viscando [7] provides traffic counts, classification, and flow analysis. The system is based on a stereo camera and coupled with a web-based interface for presenting the data.

- *DataFromSky* [4] uses aerial video from a drone to obtain an excellent top-view of the road with very few inter-object occlusions. The drone videos are analyzed off-line, providing detections, trajectories, and classifications of road users. The end user may inspect the data by using a desktop interface. Because of the physical limitations of the drone, the observational period is limited to four hours.

We have placed the frameworks listed above on a two-dimensional coordinate system shown in Figure 3.2 where each framework is placed accordingly to the relative accessibility and level of automation of the system. We define the accessibility as the ease of use for traffic practitioners without any prior knowledge in computer vision or data science. The level of automation is defined as the sophistication of the system, i.e. how close the framework is to the overall goal of fully automated traffic surveillance, including automatic

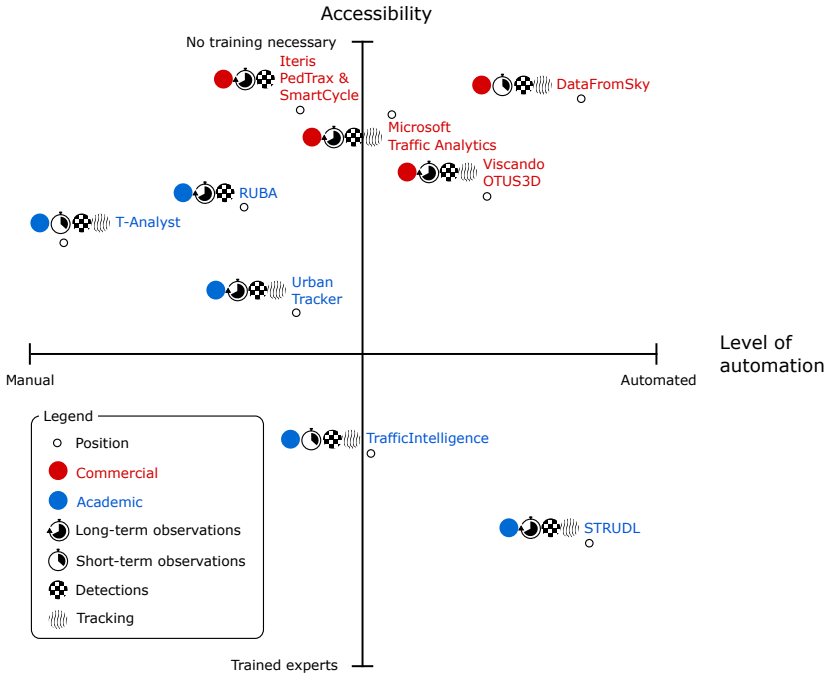


Fig. 3.2: Existing solutions for video-based traffic analysis. Solutions are positioned in the graph according to the position of the black circle. The period for which an observation might be conducted continuously is divided into two groups: short-term and long-term. Short-term observations are defined as lesser than one day whereas long-term observations consequently last more than one day.

detection, classification, and tracking of objects. As seen in the Figure, both DataFromSky and STRUDL have high scores on the level of automation. However, both tools need some manual supervision to function. STRUDL requires the annotation of 300-400 objects for each scene whereas DataFromSky uses an undisclosed process which might also include manual annotation and corrections to the final detections and tracks. It is also apparent from the diagram that all commercial solutions have high scores on the accessibility parameter whereas the academic approaches have relatively lower scores. This is a consequence of the fact that the commercial services hide the complexity of their solutions and that the fine-tuning of parameters is included in the services as well. The academic solutions provide a much higher level of configurability and are delivered as-is, leaving the required adjustments to the end user.

Other commercial services worth mentioning are Gridsmart [5] which provides detection of road users based on virtual boxes placed on the road, similar to the services from Iteris. Miovision [6] provides detection from static

cameras but also sells a portable solution for short-term observations. They provide a CNN-based object detection and classification scheme similar to the offering from Microsoft.

3 Contributions

The research within robust traffic analysis has been conducted as part of a greater research project sponsored by the European Union[1]. The research project is named *In-Depth Understanding of Accident Causation for Vulnerable Road Users* (InDeV) and involves traffic researchers, computer vision researchers, and traffic management consultants. The daily life within this project involved interdisciplinary and inter-institutional collaboration on work packages, deliverables, deadlines, and status meetings. The articles included on robust traffic analysis in this thesis reflect the diverse range of tasks conducted within the InDeV project.

We have grouped the work into three subcategories: pre-processing the data, post-processing the data, and collecting the data. The categorization is illustrated in Figure 3.3 which also shows the relationship between datasets, articles, and the programs developed within this thesis. Below, we will describe our contributions within the subcategories.

3.1 Pre-processing: The influence of the weather

In the work within pre-processing, we have investigated the influence of the weather on automated traffic surveillance and more specifically, how bad weather affects the ability of computer vision algorithms to detect road users from surveillance video. We have narrowed down our focus to the influence of rainfall and snowfall and investigated the use of pre-processing to correct for those phenomena such that the road users are easier to detect, classify, and track. When it comes to rain and snow, such pre-processing algorithms are known as rain removal or snow removal algorithms which are designed to artificially remove the apparent rain or snow in an image or video.

We have surveyed the existing rain removal algorithms and made our findings available in what we believe is the largest and most comprehensive survey on rain removal algorithms currently available. Of the surveyed rain removal algorithms, we have selected six algorithms for evaluation on the AAU RainSnow dataset. This dataset contains 21 five-minute sequences and one four-minute sequence on seven different traffic intersections. The sequences contain synchronized footage from both a visual camera and a thermal camera. For each sequence, 100 frames are randomly selected and every road user

[1]This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 635895.

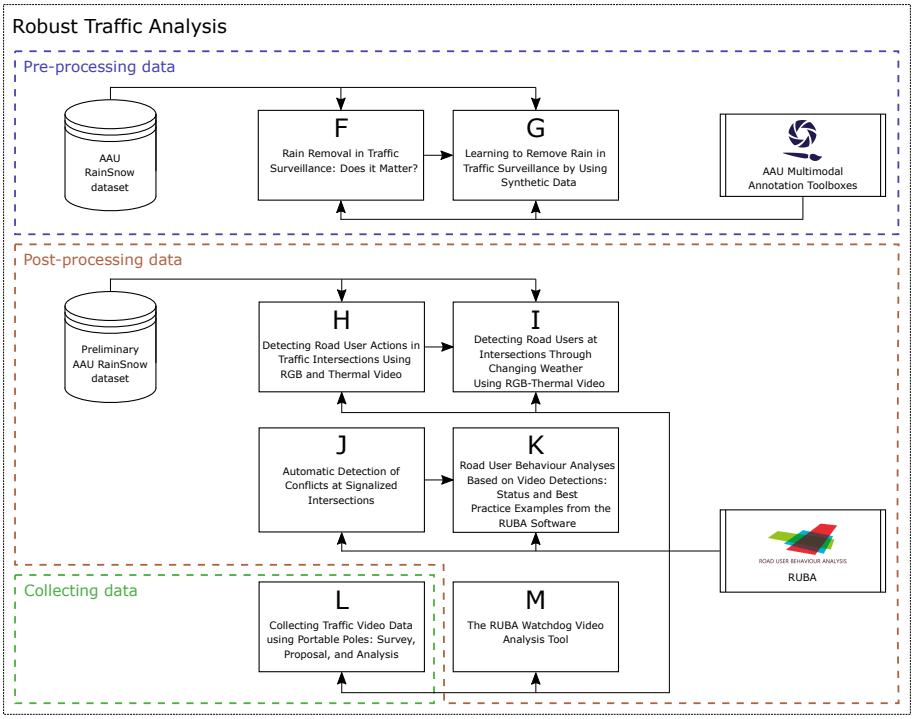


Fig. 3.3: The structure of the work within robust traffic analysis of this thesis. Boxes represents papers and reports. The letters within the boxes refer to the corresponding position in the appendix of this thesis.

within these frames is manually annotated on a pixel level. The dataset is publicly available on Kaggle[2] and is to our knowledge the world's largest publicly available dataset for thermal-visible traffic surveillance.

The six rain removal algorithms are applied on every image in the AAU RainSnow dataset and we have investigated if these rain-removed images will improve the performance of subsequent background subtraction algorithms, feature-tracking algorithms, and instance segmentation algorithms when compared to the original input images. In Figure 3.4, a sample image is de-rained using the six evaluated rain removal algorithms. For background subtraction, we applied the classical Mixture of Gaussians method [51] as well as a state-of-the-art method based on LBP and spatial diffusion [45]. For instance segmentation, we applied two methods based on the results of the COCO Detection Challenge [36]. The performance of the background subtraction and instance segmentation algorithms were evaluated relatively to the annotations of the AAU RainSnow dataset. For evaluating feature tracking performance, we tracked Lukas-Kanade features [37] for 12 seconds and then reversed the playback such that the video stopped at the point in time when the features were instantiated. If the feature-tracking was confused by the spatio-temporal raindrops or other phenomena in the videos, the feature points tracked forwards and backwards may not return to their original position. We evaluated if the forward-backward tracking was improved on the rain-removed video. The survey of rain removal algorithms, the AAU RainSnow dataset, and the evaluation described above is included in Appendix F.

Our research into rain removal algorithm led to the finding that the training of these algorithms was based on image pairs where synthetic rain was overlaid on rain-free images. The rain-free images included both outdoor and indoor images, and the visual result resembled that the rain was only falling in front of the camera. As real rain falls in the entirety of the scene, our hypothesis was that the lack of suitable training data was restraining the capabilities of the rain removal algorithms.

In 2017, I was a visiting student at the Advanced Driver Assistance Systems research group at Universitat Autònoma de Barcelona who were creating a virtual world [41] for teaching self-driving cars how to drive. In such a virtual world, it is possible to simulate two scenes where the only difference is the addition of virtual rain. Contrary to the addition of synthetic rain on real images, synthetic rain on synthetic images fall in the entirety of the scene, resembling real rain. With the kind assistance of the research group in Barcelona, we created video sequences from four different scenes in the virtual world. We used the image pairs of rain free and rain affected images to train a CNN to convert rain images into rain-free images.

[2] `https://www.kaggle.com/aalborguniversity/aau-rainsnow`

(a) Original **(b)** Median **(c)** Garg2007 [23] **(d)** Kang2012 [30]

(e) Kim2015 [31] **(f)** Fu2017 [22] **(g)** Zhang2017 [50]

Fig. 3.4: Sample results from the evaluated rain removal algorithms on the AAU RainSnow dataset.

The performance of our proposed rain removal algorithm was both evaluated on the traditional signal processing metrics such as Peak Signal-to-Noise Ratio but also on the ability to improve the subsequent performance of an object detection algorithm on the AAU RainSnow dataset. The results showed that it is indeed difficult to provide an efficient training set for rain removal algorithms. However, given that we were the first to use a fully synthetic dataset for training rain removal algorithms, we have given directions for others to improve on. The work on synthetic rain removal is included in Appendix G.

3.2 Post-processing: Robustness of use

When we started the work on robust traffic analysis in late 2013, there was a disconnection between the video analysis tools that were available to experts with a background in computer science and the tools that were available to the traffic researchers. If we relate to the diagram of Figure 3.2, the only academic frameworks available at the time were TrafficIntelligence [26] and T-Analyst [8]. TrafficIntelligence is a command-line tool that requires a basic understanding of computer vision whereas T-Analyst requires that the traffic researcher manually inspects the video for events of interest.

At the time, we decided to build a tool that could bridge the gap between

Fig. 3.5: A sample use-case of the RUBA software. Two detectors are defined to find possible encounters between cyclists and right-turning vehicles.

manual video analysis and the command-line tools that are available to computer scientists. We aimed for a watch-dog that would look for situations of interest in the traffic videos and that the tool should be accessible to traffic researchers and practitioners. The result was RUBA, a program that combined off-the-shelf computer vision algorithms with a graphical user interface. A traffic researcher may use RUBA to detect encounters between different road users, for instance between vehicles and vulnerable road users. Once RUBA has finished processing, the researcher will look at the events that RUBA has marked as interesting and decide if they should be discarded or looked further into, for instance in T-Analyst. A sample detection is shown in Figure 3.5. RUBA is currently used for research and teaching purposes at the traffic research group at AAU and has been used by other universities as well.

RUBA is open-souce and publicly available for Windows and MacOS on Bitbucket[3] where a comprehensive user manual is available. This manual is included in Appendix M. The preliminary version of RUBA, denoted as TrafficDetector, was presented at a workshop for traffic researchers and practitioners in 2014 and summarized in the abstract located in Appendix J. The applications of a more recent version of RUBA were presented in 2017. This work is included in Appendix K.

We have also experimented with RUBA for counting road user actions at intersections. In the work presented in Appendix H, we selected a number of pre-defined paths that defined the actions of road users turning right and left and cyclists going straight in an intersection. If a road user enters the pre-defined zones in a certain order, we incremented the corresponding counter. Detection was performed separately using visual and thermal video

[3] `https://bitbucket.org/aauvap/ruba/`

and fused at decision-level. We later expanded the evaluation to 16 hours of visual and thermal video captured under different weather and illumination conditions and compared the results to the feature-based TrafficIntelligence framework [26]. The results showed that the TrafficIntelligence gave higher precision but similar recall when compared to RUBA. The extended evaluation is included in Appendix I.

3.3 Collecting data

At the beginning of the InDeV project, we were gathering knowledge on how to collect video data from urban roads. We surveyed a range of existing options for temporary installations, ranging from lightweight solutions carried by a human to a heavyweight installation that could only be moved by a truck. The work, which is included in Appendix L, also describes the development of a portable platform for temporary video recording that was developed by my colleagues based on the findings of our survey.

4 Sub-conclusion

Our main contributions within robust traffic analysis have been the following:

- A detailed study and evaluation of the influence of rain and snow in traffic surveillance.
- A comprehensive survey on rain removal algorithms.
- The publicly available AAU RainSnow dataset containing traffic surveillance video captured by visual and thermal cameras.
- A rain removal algorithm trained on fully synthetic data.
- The development of the RUBA software that enables the use of computer-vision-aided traffic analysis to traffic researchers with no prior knowledge of computer science.

References

[1] "Pedtrax," https://www.iteris.com/system/files/content/resource/2017-09/PedTrax_Datasheet_Rev2_Aug2017.pdf, year=2017.

[2] "Section control: Monitoring the average speed on a specific road section," https://www.jenoptik.com/products/traffic-safety-systems/section-control, accessed: 2018-11-03.

[3] "Tollchecker flexible free-flow tolling solutions," https://www.vitronic.com/fileadmin/user_upload/Verkehrstechnik/Downloads/Englisch/Brochures/VITRONIC-TOLLCHECKER-Flexible-Free-Flow-Tolling-Solutions.pdf, 2016.

[4] "Datafromsky," http://datafromsky.com/, 2018, accessed: 2018-11-05.

[5] "The gridsmart system - gridsmart," https://gridsmart.com/products/gridsmart-solution/, note = Accessed: 2018-11-08, 2018.

[6] "Multimodal detection," https://www2.miovision.com/multimodal-detection-overview, 2018, accessed: 2018-11-08.

[7] "Otus3d," https://viscando.com/en/otus3d, 2018, accessed: 2018-11-05.

[8] "T-analyst software for semi-automated video processing," www.tft.lth.se/software, 2018, accessed: 2018-11-05.

[9] N. Agerholm, C. Tønning, T. K. O. Madsen, C. H. Bahnsen, T. B. Moeslund, and H. S. Lahrmann, "Road user behaviour analyses based on video detections: Status and best practice examples from the ruba software," in *24th ITS World Congress Montreal 2017*, 2017, pp. 1–10.

[10] M. Ahrnbom, K. Astrom, and M. Nilsson, "Fast classification of empty and occupied parking spaces using integral channel features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 9–15.

[11] T. Alldieck, C. H. Bahnsen, and T. B. Moeslund, "Context-aware fusion of rgb and thermal imagery for traffic monitoring," *Sensors*, vol. 16, no. 11, p. 1947, 2016.

[12] G. Ananthanarayanan, P. Bahl, P. Bodík, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha, "Real-time video analytics: The killer app for edge computing," *computer*, vol. 50, no. 10, pp. 58–67, 2017.

[13] L. O. Andrews Sobral, L. Schnitman, and F. De Souza, "Highway traffic congestion classification using holistic properties," in *10th IASTED International Conference on Signal Processing, Pattern Recognition and Applications*, 2013.

[14] R. Babari, N. Hautière, É. Dumont, N. Paparoditis, and J. Misener, "Visibility monitoring using conventional roadside cameras–emerging applications," *Transportation research part C: emerging technologies*, vol. 22, pp. 17–28, 2012.

[15] C. H. Bahnsen and T. B. Moeslund, "Rain removal in traffic surveillance: Does it matter?" *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–18, 2018.

[16] N. Buch, S. A. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 920–939, 2011.

[17] C. C. Chen, "Attenuation of electromagnetic radiation by haze, fog, clouds, and rain," RAND CORP SANTA MONICA CA, Tech. Rep., 1975.

[18] S. R. E. Datondji, Y. Dupuis, P. Subirats, and P. Vasseur, "A survey of vision-based traffic monitoring of road intersections," *IEEE transactions on intelligent transportation systems*, vol. 17, no. 10, pp. 2681–2698, 2016.

[19] P. Duthon, F. Bernardin, F. Chausse, and M. Colomb, "Benchmark for the robustness of image features in rainy conditions," *Machine Vision and Applications*, pp. 1–13, 2018.

[20] European Road Safety Observatory, "Traffic safety basic facts 2017 - main figures," *Brussels: European Commission, Directorate General for Transport*, 2017.

[21] Q. Fan, J. Yang, G. Hua, B. Chen, and D. P. Wipf, "A generic deep architecture for single image reflection removal and image smoothing." in *ICCV*, 2017, pp. 3258–3267.

[22] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *Computer Vision and Pattern Recognition, IEEE Conference on*, July 2017.

[23] K. Garg and S. K. Nayar, "Vision and rain," *International Journal of Computer Vision*, vol. 75, no. 1, p. 3, 2007.

[24] J. C. V. Guerra, Z. Khanam, S. Ehsan, R. Stolkin, and K. McDonald-Maier, "Weather classification: A new multi-class dataset, data augmentation approach and comprehensive evaluations of convolutional neural networks," *arXiv preprint arXiv:1808.00588*, 2018.

[25] C. Hydén, "The development of a method for traffic safety evaluation: The swedish traffic conflicts technique," *BULLETIN LUND INSTITUTE OF TECHNOLOGY, DEPARTMENT*, no. 70, 1987.

[26] S. Jackson, L. F. Miranda-Moreno, P. St-Aubin, and N. Saunier, "Flexible, mobile video camera system and open source video analysis software for road safety and behavioral analysis," *Transportation research record*, vol. 2365, no. 1, pp. 90–98, 2013.

[27] M. B. Jensen, M. Ahrnbom, M. Kruithof, K. Åström, M. Nilsson, H. Ardö, A. Laureshyn, C. Johnsson, and T. B. Moeslund, "A framework for automated traffic safety analysis from video using modern computer vision," in *Transportation Research Board (TRB) 98th Annual Meeting*, 2018.

[28] J.-P. Jodoin, G.-A. Bilodeau, and N. Saunier, "Urban tracker: Multiple object tracking in urban mixed traffic," in *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*. IEEE, 2014, pp. 885–892.

[29] P. Jonsson, "Road condition discrimination using weather data and camera images," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 1616–1621.

[30] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *Image Processing, IEEE Transactions on*, vol. 21, no. 4, pp. 1742–1755, 2012.

[31] J.-H. Kim, J.-Y. Sim, and C.-S. Kim, "Video deraining and desnowing using temporal correlation and low-rank matrix completion," *Image Processing, IEEE Transactions on*, vol. 24, no. 9, pp. 2658–2670, 2015.

[32] S. Lee, S. Yun, J.-H. Nam, C. S. Won, and S.-W. Jung, "A review on dark channel prior based image dehazing algorithms," *EURASIP Journal on Image and Video Processing*, vol. 2016, no. 1, p. 4, 2016.

[33] Q. Li, E. A. Bernal, M. Shreve, and R. P. Loce, "Scene-independent feature-and classifier-based vehicle headlight and shadow removal in video sequences," in *2016 IEEE Winter Applications of Computer Vision Workshops (WACVW)*. IEEE, 2016, pp. 1–8.

[34] Y. Li, S. You, M. S. Brown, and R. T. Tan, "Haze visibility enhancement: A survey and quantitative benchmarking," *Computer Vision and Image Understanding*, vol. 165, pp. 1–16, 2017.

[35] F.-J. Lin and T.-P. Wang, "Metric learning for weather image classification," *Multimedia Tools and Applications*, pp. 1–13, 2018.

[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[37] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," 1981.

[38] Z. Luo, B. Frederic, C. Lemaire, J. Konrad, S. Li, A. Mishra, A. Achkar, J. Eichel, P.-M. Jodoin *et al.*, "Mio-tcd: A new benchmark dataset for vehicle classification and localization," *IEEE Transactions on Image Processing*, 2018.

[39] S. Lyu, M.-C. Chang, D. Du, L. Wen, H. Qi, Y. Li, Y. Wei, L. Ke, T. Hu, M. Del Coco *et al.*, "Ua-detrac 2017: Report of avss2017 & iwt4s challenge on advanced traffic monitoring," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2017, pp. 1–7.

[40] M. Naphade, D. C. Anastasiu, A. Sharma, V. Jagrlamudi, H. Jeon, K. Liu, M.-C. Chang, S. Lyu, and Z. Gao, "The nvidia ai city challenge," *IEEE Smart-World, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (Smart-World/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 2017.

[41] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3234–3243.

[42] A. Sanin, C. Sanderson, and B. C. Lovell, "Shadow detection: A survey and comparative evaluation of recent methods," *Pattern recognition*, vol. 45, no. 4, pp. 1684–1695, 2012.

[43] E. P. Shettle, "Models of aerosols, clouds, and precipitation for atmospheric propagation studies," in *In AGARD, Atmospheric Propagation in the UV, Visible, IR, and MM-Wave Region and Related Systems Aspects 14 p (SEE N90-21907 15-32)*, 1990.

[44] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[45] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "Subsense: A universal change detection method with local adaptive sensitivity," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 359–373, 2015.

[46] Å. Svensson and C. Hydén, "Estimating the severity of safety related behaviour," *Accident Analysis & Prevention*, vol. 38, no. 2, pp. 379–385, 2006.

[47] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 3–19, 2014.

[48] M. Vollmer and K.-P. Möllmann, *Infrared thermal imaging: fundamentals, research and applications*. John Wiley & Sons, 2010.

[49] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "Cdnet 2014: an expanded change detection benchmark dataset," in *2014 IEEE conference on computer vision and pattern recognition workshops*. IEEE, 2014, pp. 393–400.

[50] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *Computer Vision and Pattern Recognition, IEEE Conference on*, 2017.

[51] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2. IEEE, 2004, pp. 28–31.

Chapter 4

Conclusion

This PhD covers work conducted from 2013 to 2018 within two overall themes: multi-modal analysis and robust traffic analysis

Within multi-modal analysis, we investigated how to obtain synchronized imagery from visual, depth, and thermal sensors and how to effectively register a spatial position in one modality to another. We used our tri-modal acquisition and registration platform to conduct research within people re-identification and people segmentation where we experimented with different detection and description methods. As part of this work, we released a publicly available tri-modal dataset for people segmentation. The dataset contains annotations of people on a pixel level and the disparate nature of the three modalities meant that a new tool was required. This annotation tool has been extended as part of this PhD and has subsequently been used for many other research projects in our laboratory. When the modalities are registered, one should know how to use the information from the sensors. In traffic surveillance, we have investigated how to combine information from visual and thermal cameras using both the quality of the images and the related contextual information.

In the work within robust traffic analysis, we have studied the influence of the weather on the visual domain. More specifically, we have investigated the influence of rainfall and snowfall on visual traffic surveillance conducted in urban areas. We surveyed the field of rain removal algorithms, whose promise is to mitigate the visual influence of the spatio-temporal streaks by pre-processing the input image or video. Six rain removal algorithms were selected in order to investigate if the removal of rain from traffic surveillance video would increase the performance of subsequent background subtraction, instance segmentation, and feature tracking algorithms. For evaluation, we collected and annotated a new dataset consisting of visual-thermal video from seven Danish intersections, all featuring rain or snow. The dataset

is so far the world's largest publicly available collection of visual-thermal images for surveillance. Results showed improvements in feature tracking and background subtraction but did not provide improvements in instance segmentation.

The survey revealed that many rain removal algorithms were trained on rain-free images overlaid with synthetic rain streaks. These images do well in simulating the rain in front of the camera but fails to model the presence of rain anywhere else in the scene. In order to account for this, we investigated the use of fully synthetic sequences from a virtual world where rain is rendered in the entirety of the scene. Using these data, we trained a rain removal algorithm and compared it with the state-of-the-art. The results showed that it is indeed difficult to find a proper procedure for training rain removal algorithms.

In traffic safety analysis, traffic researchers are analyzing the behavior of road users in order to understand the causation of accidents. Traditionally, this has been accomplished by filming the road and manually traversing the video for interesting situations. However, this is very time-consuming when the situations of interest are sparse. In order to reduce the amount of video for manual inspection, we have developed the RUBA software tool. The traffic researchers use RUBA to automatically detect interactions between road users that are subsequently selected for manual review. The traffic researchers at AAU as well as other institutions have picked up the use of RUBA for long-term video analysis.

We have tested RUBA for detection and counting of turning road user movements at urban intersections and compared it against a more advanced, feature-based tracker. The results showed that the feature-based tracker was more precise but that both systems produced similar recall rates, which is important for a watch-dog system. In extension of our work within traffic surveillance, we investigated several options for establishing a portable video acquisition platform, which led to the construction of a new portable pole.

As part of the dissemination activities, we have authored an article on deep learning for the Danish magazine on popular science, *Aktuel Naturvidenskab*.

The research conducted within this PhD leaves several directions for others to follow. In multi-modal analysis, there are still open questions on how to combine and use information from multi-modal sensors. In traffic analysis, the advent of deep learning has made detection and tracking of road users a commodity under the premise of few occlusions and good weather conditions. However, in complex scenes, bad weather, or insufficient lighting, the current techniques break down. In order to combat this, we must gather new insights, new datasets, and novel ways to automatically adapt to changing conditions.

Part II

Multi-Modal Analysis

Paper A

Tri-modal Person Re-Identification with RGB, Depth and Thermal Features

Andreas Møgelmose, Chris Bahnsen, Thomas B. Moeslund

The paper has been published in the
Proceedings of the 9th Workshop on Perception Beyond the Visible Spectrum, CVPR Workshops, pp. 301-307, 2013.

The layout has been revised.

Abstract

Person re-identification is about recognizing people who have passed by a sensor earlier. Previous work is mainly based on RGB data, but in this work we for the first time present a system where we combine RGB, depth, and thermal data for re-identification purposes. First, from each of the three modalities, we obtain some particular features: from RGB data, we model color information from different regions of the body; from depth data, we compute different soft body biometrics; and from thermal data, we extract local structural information. Then, the three information types are combined in a joined classifier. The tri-modal system is evaluated on a new RGB-D-T dataset, showing successful results in re-identification scenarios.

1 Introduction

Person re-identification is about recognizing people who have passed by a sensor earlier. It is useful in many places where it is desirable to obtain knowledge of the flow of people: airports, transit centers, shopping malls, amusement parks, etc. It can either be knowledge of a single person's movement, or movement patterns in general by combining the patterns of many people. In some cases it is possible to set up a system, which is able to view the entire scene, as in [16,19]. However, in indoor scenes it is often not feasible to place one camera with a full overview. This is where re-identification enters play. It allows the system designer to place sensors at certain bottlenecks and identify people when they pass these.

Re-identification has the specific distinction from e.g. biometric access control systems that it must be able to enroll new people on-the-fly and without their specific collaboration. On the other hand, the recognition performance does not necessarily have to be as strong as in access control systems, since re-identification systems are more concerned with the general trend of movement as opposed to the movement of each individual.

Re-identification has been an active research area for the past decade, but almost exclusively focused on standard RGB-data. This makes sense since many venues have a large network of already installed RGB surveillance cameras. However, as new and more advanced sensor types become cheaply available, we believe it is time to extend the work to multiple modalities. This is the exact focus of this work, where we present a novel approach that integrates RGB, depth, and thermal data in a re-identification system. An example of RGB, depth, and thermal images for a subject in our dataset is shown in Figure A.5.

This paper is structured as follows: Section 2 briefly covers the existing work done on the topic of re-identification, with special focus on the few multi-modal and/or non-RGB-based contributions. Section 3 describes how

the inputs from the three modalities are aligned. In sections 4 and 5, the features and re-identification methods are presented. Section 6 shows the dataset and covers the results our system achieves on it. Finally, section 7 concludes the paper.

2 Related work

In [6] soft-biometrics based on RGB data are used to track people across different cameras. Both body and facial soft biometrics are extracted and combined in the final system. The body soft biometrics are all related to color: hair, skin, upper, and lower body clothing. In [7] the notion of tracking people across a multi-camera setup is also followed. Different soft biometric features are reviewed and discussed in the context of re-identification. A part-based appearance approach is found to perform the best, but being sensitive to how the object is divided into parts. In [8] each person is also divided into parts from which features are extracted. The division is here based on finding symmetry axes and the soft biometric features are color histograms, stable color regions and highly structured patches that reoccur. A division is also applied in [10] using similar features. A boosting approach is then introduced to select the most discriminative features. In [1] a similar idea is proposed, i.e., a more reliable classification can be obtained if only the most discriminative features are used for each image region. Moreover they model the uncertainties (covariances) of each feature to improve their results. In [22] a person is divided into six horizontal stripes where each is described in terms of color and texture. The novelty of the work if the formulation of the re-identification problem as a matter of learning the optimal distance measure that minimizes the probability of miss-classification.

All of the above approaches are based on RGB data. Using multi-modal sensing in re-identification is a very new concept and so far only a few works have been reported. In [20] a two-stage recognition approach is followed. First soft-biometrics based on depth data are extracted and secondly RGB data are used in the final classification step. The depth-based soft biometrics are anthropometric measurements and estimated manually. The key finding is that soft biometrics can be used as a pruning step in a recognition system. While this is very interesting, the introduction of manual measurements is not desirable for an automatic re-identification system. In [2] a re-identification method based solely on depth features is presented. The work uses several normalized measures of body parts, calculated from joint positions. Measures of the body's "roundness", which roughly estimates the volume of the torso, are included. High depth resolution is required for this to work and hence it is only suitable when subjects are close to the sensor. The paper is focused solely on the re-identification step and does not treat identification or extraction of

joints. In [13] thermal data are used in a re-identification system. The work expands the work reported in [12] where SIFT features are used to model each person. They work on gait data from a side view and can thou track each body part reliably. From each of these a codebook signature is learned over time and combined with the spatial feature distribution found using an Implicit Shape Model.

As opposed to the works described above, in this paper we introduce a truly multi-modal approach based on RGB, depth and thermal data. Moreover, our system is fully automated both in terms of feature extraction, but also when it comes to enrollment.

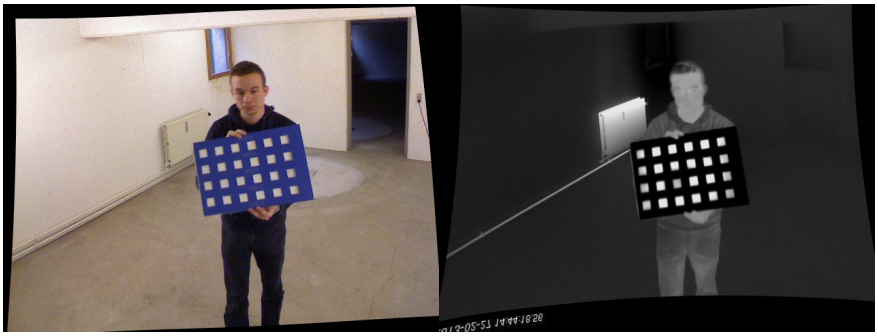
3 Registration

Since no sensor is able to capture all three modalities at once, a registration of the inputs must take place allowing to map a specific point from one modality to the others. In this work, the Microsoft® Kinect™ for XBOX360 has been used to capture RGB and depth data. A thermal camera (AXIS Q1922) was mounted straight over the Kinect's RGB camera lens with a distance between the lens centers of 70 mm. For registering the tri-modal imagery of this work, we need only to register images from the thermal and visual modalities, as the Kinect provides a factory calibrated registration between the RGB and depth data.

Traditional image registration techniques used for spatially aligning stereo imagery cannot be directly applied to the thermal-visible domain due to the fundamental physical differences of the two modalities, thus rendering the process of finding corresponding features in both imagery is unfeasible. In our setup, objects appear at distances between 1 and 4 meters from the cameras, which makes methods like infinite homography and stereo geometric unusable [14]. Instead we first use stereo rectification to transform the epipolar lines to lines parallel with either the x or y axis [9]. This reduces the search for corresponding points to one dimension. Next we apply the notion that the distance between corresponding points in the two images is inversely proportional to the depth of the points if the cameras are only translated with respect to each other [9]. Since the epipolar lines are transformed to lie along the image scanlines, the disparity between corresponding points will lie mainly either on the x or y axes, and we may thus find the relationship between the inverted depth and the induced disparity and use this property for rectifying the images.

The stereo calibration requires the knowledge of the intrinsic and extrinsic camera parameters of both cameras. In order to determine these, we use the calibration board proposed by [21] with an A3-sized cut-out checkerboard and a heated plate as a viable backdrop. By using standard camera calibration

and stereo geometric tools we are able to rectify both images as seen in Figure A.1.



(a) **(b)**

Fig. A.1: Stereo rectified multimodal imagery in the (a) RGB and (b) thermal domains.

We used 34 image pairs of the calibration board distributed throughout the entire scene for the calibration of the cameras. For each corner of the chessboard in each image, we extract the corresponding depth. The configuration of cameras placed vertically implies that the disparity of the points in the rectified image lies mainly on the x-axis. Therefore, we use a robust curve fitting tool to find a linear regression that fits the disparity in the x-direction as a function of the inverted distance in the z-direction. The regression is computed off-line for all calibration points and stored for online lookup of the displacement. The result of this procedure is a direct pixel-to-pixel correspondence between the different images.

4 Multi-modal features

The proposed system uses a combination of RGB, depth, and thermal features to perform the re-identification task. This section explains how the feature extraction is performed for each modality. Before the extraction, the subject must first be located at pixel level. The foreground segmentation of the subject is performed on the depth image by means of Random Forest [18]. This process is performed computing random offsets of depth features as follows:

$$f_\theta(\mathcal{D}, \mathbf{x}) = \mathcal{D}_{\left(\mathbf{x}+\frac{\mathbf{u}}{\mathcal{D}_\mathbf{x}}\right)} - \mathcal{D}_{\left(\mathbf{x}+\frac{\mathbf{v}}{\mathcal{D}_\mathbf{x}}\right)}, \tag{A.1}$$

where $\theta = (\mathbf{u}, \mathbf{v})$, and $\mathbf{u}, \mathbf{v} \in \mathbb{R}^\mathbf{2}$ is a pair of offsets, depth invariant. Thus, each θ determines two new pixels relative to $\mathbf{x}$, the depth difference of which accounts for the value of $f_\theta(\mathcal{D}, \mathbf{x})$. Using this set of random depth features,

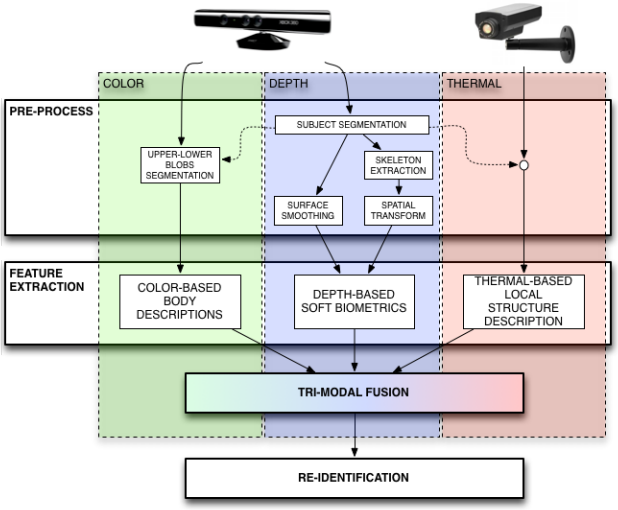


Fig. A.2: Pipeline of the proposed tri-modal re-identification system.

Random Forest is trained for a set of trees, where each tree consists of split and leaf nodes (the root is also a split node). Finally, a final pixel probability of body part membership l_i is obtained as follows:

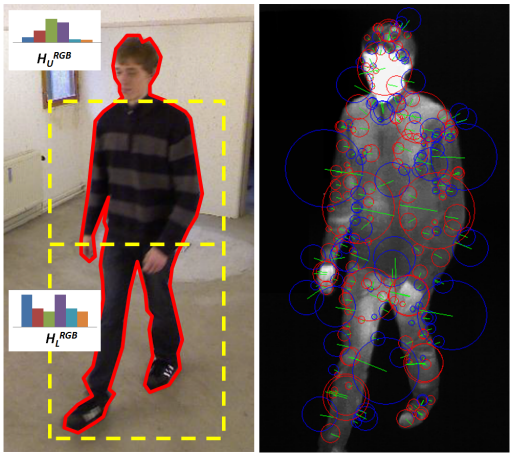
$$P\left(l_i|\mathcal{D},\mathbf{x}\right) = \frac{1}{\tau}\sum_{j=1}^{\tau} P_j\left(l_i|\mathcal{D},\mathbf{x}\right), \tag{A.2}$$

where $P\left(l_i|\mathcal{D},\mathbf{x}\right)$ is the PDF stored at the leaf, reached by the pixel for classification $(\mathcal{D},\mathbf{x})$ and traced through the tree j, $j \in \tau$. After this process, the foreground segmentation mask of the subject is transformed to the coordinate system in the two other modalities, and the features are extracted.

The system uses multi-shot person models. Thus, a person is not modeled based on only one frame, but on all frames in a pass. A pass is defined as the act of entering the frame, walking by the camera, and exiting it. In our dataset only one person is present at a time, so no tracking is necessary. Next, we describe how the features from each modality are described and fused in order to perform the on-line re-identification task. Figure A.2 summarizes the main modules, modalities and strategies considered in the proposed re-identification system.

4.1 RGB features

After foreground segmentation is performed, the features that are used for the RGB modality are color histograms in two parts, as shown in Figure A.3(a). One histogram H_U^{RGB} is derived from the upper body, one H_L^{RGB} from the



(a) (b)

Fig. A.3: (a) Histograms of RGB color distributions for upper body H_U^{RGB} and lower body H_L^{RGB} parts of the subject. (b) Detected SURF keypoints on the thermal modality.

lower. This is done for each frame in which the subject is detected. A histogram of 20 bins is created for each channel, for a total of 60 bins per body part. Thus, in total the RGB feature vector has 120 dimensions, and one is created per frame. After a pass ends, the histograms are averaged, and the final feature vector is the mean across the frames.

4.2 Depth features

Given an input depth frame containing a subject (Figure A.4(a)), and once the pixel-ground segmentation of the subject into body parts is performed, the skeleton is also extracted applying Mean Shift [18] (Figure A.4(b)). Since our dataset contains only raw images, the built-in skeleton-extraction from the Kinect could not be used. Then, the subject point cloud is spatially transformed in order to align the skeleton with the camera frame coordinate system by means of an affine three-dimensional transformation of the point cloud (Figure A.4(c)). Note that because of the 3D transformation we loose some information of the body surface due to the lack of information inherent to the viewpoint. Thus, the noisy subject's surface is smoothed (Moving Least Squares surface reconstruction method) and up-sampled to fill the holes (Figure A.4(d)). Now we can compute soft biometrics from the corrected 3D skeleton and the 3D surface of the aligned body, which can be then inversely transformed to return to the original space and estimate real measurements of the body. From a given depth frame $\mathcal{D}_i$, information invariant to the rotation of the subject with respect to the camera viewpoint can now be extracted. In particular, we have estimated three sets of soft biometrics:

Frontal curve model: The model encodes the distances from the points in subject's surface (transformed and smoothed, as seen in Figure A.4(d)) to their corresponding projection line, either head-to-neck or neck-to-torso line. These distances in millimeters are encoded in a real-valued vector $\mathbf{f}_i$, resampled to size 150 and equalized for normalization purposes (Figure A.4(e)).

Thoracic geodesic distances: Corresponds to the vector $\mathbf{g}_i$. It contains the length of lines on the body surface from one side of the body to the other. The area in which these are found is the trapezoid defined by left shoulder, right shoulder, right hip, and left hip, and each entry of $\mathbf{g}_i$ contains the geodesic distance in millimeters of a horizontal line in the trapezoid projected to the surface of the torso. $\mathbf{g}_i$ is resampled to size 90 (Figure A.4(f)).

Anthropometric relations: Given the extracted body skeleton, the lengths of 7 inter-joint segments connecting the body parts, as shown in Figure A.4(c), are computed and stored as $\mathbf{a}_i$.

Thus, the vector representing the set of depth features for a subject in the scene at a particular depth frame $\mathcal{D}_i$ is defined as:

$$\boldsymbol{\delta}_i = \{\mathbf{f}_i, \mathbf{g}_i, \mathbf{a}_i\},$$

where $|\boldsymbol{\delta}_i| = 247$. Finally, the vector describing the subject pass $D = \{F, G, A\}$ is computed by averaging the set of the standardized frame-level depth feature vectors $\{\boldsymbol{\delta}_1, ..., \boldsymbol{\delta}_N\}$ as:

$$D = \frac{1}{N} \sum_{j \in N} \frac{\boldsymbol{\delta}_j - \bar{\boldsymbol{\delta}}}{\boldsymbol{\sigma}_{\boldsymbol{\delta}}}, \tag{A.3}$$

where $|D| = 247$, and $\bar{\boldsymbol{\delta}}$ and $\boldsymbol{\sigma}_{\boldsymbol{\delta}}$ correspond to the mean depth vector and the vector of the standard deviations, respectively. Moreover, as a previous step to this computation and due to the noisy nature of the captured depth data (clothes deformation, waving arms in front of the torso, and so forth), the possible outliers are detected and discarded in each $\boldsymbol{\delta}_i$. This step consists also in standardizing the set of depth feature vectors but to a modified Z-score [11] and discarding those values higher than 3.5 in absolute value.

4.3 Thermal features

Since the thermal images contain no color information, the color histogram approach does not work here. Instead, SURF [3] is employed. Within the contour supplied by the detection stage, SURF-descriptors are extracted. There is no fixed number of descriptors, all that are above a certain quality threshold are extracted. A typical number is around 150 descriptors per subject per frame, depending on the contour's size and quality. As opposed to the RGB histograms there is no direct way to average the descriptors, so the model for people in the thermal modality is all SURF descriptors of the subject extracted

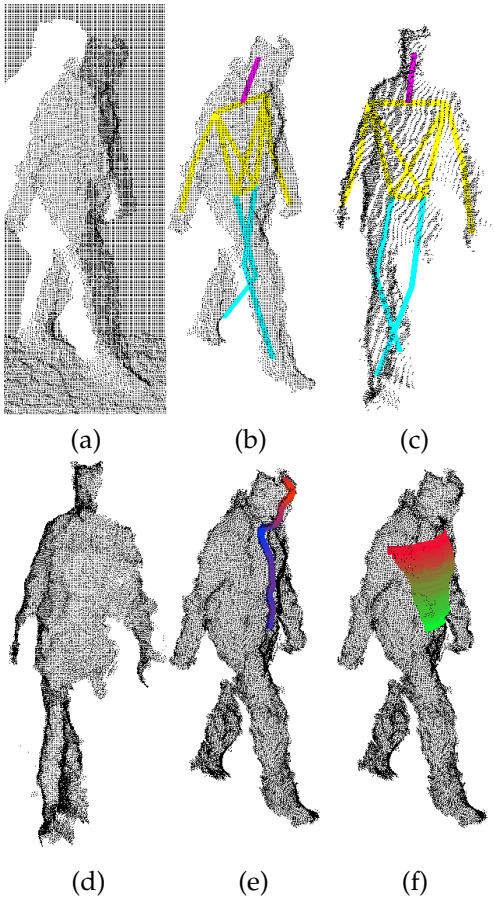


Fig. A.4: (a) The raw depth data. (b) The pixel-ground segmentation of the subject and the skeleton. (c) After aligning the skeleton with the camera frame. (d) Smoothed data. (e) Vertical projection lines. (f) Geodesic distances.

over all frames in a pass. We define the set of detected and described SURF points as S, see Figure A.3(b).

5 Re-identification

In order to perform the re-identification task, previously computed feature vectors for the three modalities have to be fused and analyzed to classify each subject. The process has two steps:

1. Determine whether the subject is a new or an already known person.
2. Do one of the following two tasks:
   (a) If known, determine the ID of the person.
   (b) If new, enroll the person.

In step 1, a comparison of the current subject with the list of known subjects is done. Taking into account that the set of known persons is built on-the-fly, for the first evaluations only a few comparisons have to be performed.

To estimate whether the subject has to be considered new or re-identified, we compute the following confidence score based on the combination of the three modalities scores:

$$C(U_1, U_2) = \alpha \cdot d_{\text{RGB}}(H_1, H_2) + \beta \cdot \frac{1}{d_{\text{depth}}(D_1, D_2)} + \\ + \gamma \cdot \frac{1}{d_{\text{thermal}}(S_1, S_2)},$$

where $U_1 = \{H_1, D_1, S_1\}$ is the set of three modality descriptors (H_1 color histograms, D_1 depth feature vectors, and S_1 SURF descriptors on the thermal data) for a user in the dataset, and $U_2 = \{H_2, D_2, S_2\}$ are the three sets of descriptors for a new test subject. Coefficients α, β, and γ assigns a proper weight to each of the three modalities scores in a late fusion fashion so that $\alpha + \beta + \gamma = 1$. The weights are static and were set based on experimentation, but for future work, and especially larger datasets, a learning approach for the weights would have to be investigated. The higher the output of $C(U_1, U_2)$, the more reliable re-identification. Because $d_{\text{depth}}(D_1, D_2)$ and $d_{\text{thermal}}(S_1, S_2)$ returns low values in case of good identifications, the reciprocal is used when fused.

For comparing two subjects in the RGB-modality, the Bhattacharyya distance [5] is used:

$$d_{\text{RGB}}(H_1, H_2) = \sqrt{1 - \sum_I \frac{\sqrt{H_1(I) H_2(I)}}{\sqrt{\sum_I H_1(I) \cdot \sum_I H_2(I)}}}, \tag{A.4}$$

where $d_{\text{RGB}}(H_1, H_2)$ describes the distance between histograms H_1 and H_2, and $H(I)$ is the value of bin I in the histogram H. The distance is a number between 0 and 1, where 0 is a perfect match.

For comparing across subjects in the depth modality $D = \{F, G, A\}$, the following similarity measure is computed:

$$\begin{aligned} d_{\text{depth}}(D_1, D_2) = W_F(1 - \exp^{-\sum_i w_i (F_1^i - F_2^i)^2}) + \\ + W_G(1 - \exp^{-\sum_j w_j (G_1^j - G_2^j)^2}) + \\ + W_A(1 - \exp^{-\sum_k w_k (A_1^k - A_2^k)^2}). \end{aligned} \tag{A.5}$$

One distance is computed for each of the three depth features, which is in the range [0..1], the lower the distance, the higher the similarity. Coefficients W_F, W_G, and W_A assigns a proper weight to each of the three types of depth feature sets so that $W_F + W_G + W_A = 1$. Moreover, individual feature weights w assign a weight to each particular depth feature value, pre-computed based on a training stage applying ReliefF [17]. In out case the variables were set to $W_F = 0.8$, $W_G = 0.1$, and $W_A = 0.1$.

In the thermal domain, the SURF-descriptors are matched against each other with no spatial information resolved. Each matched feature contributes a vote. Thus the metric is the number of votes for a specific known person across all the frames in the model:

$$d_{\text{thermal}}(S_1, S_2) = \sum_{N_{S_2}} H(n_{\text{votes}}(S_1, S_2)), \tag{A.6}$$

where $n_{\text{votes}}(S_1, S_2)$ computes the number of matches between SURF descriptors S_1 on the reference image and SURF descriptors S_2 on the test image based on Euclidean distance criterion. H refers to the Heaviside step function, ensuring that each frame in a pass can only contribute one vote, and N are the frames in the model for S_2.

5.1 Determine if new

In order to determine if a person is new, once values for α, β, and γ are established based on a cross-validation of a training stage, two thresholds, T_N and T_R are also experimentally computed. If $C < T_N$, the subject is considered new. If $C > T_R$ the subject is assigned a known ID (re-identified). Since a false positive is more serious than a false negative in re-identification, we have a buffer zone when $T_N \leq C \leq T_R$ where the system ignores the subject because we are uncertain whether it is a new person or just a bad match to an existing one. In our system we used $T_N = 6$ and $T_R = 10$, but the exact value of the thresholds seemed to be relatively flexible.

5.2 ID determination

The assignment of an ID to an already existing user for re-identification is straightforward using the confidence score C obtained from the previous step.

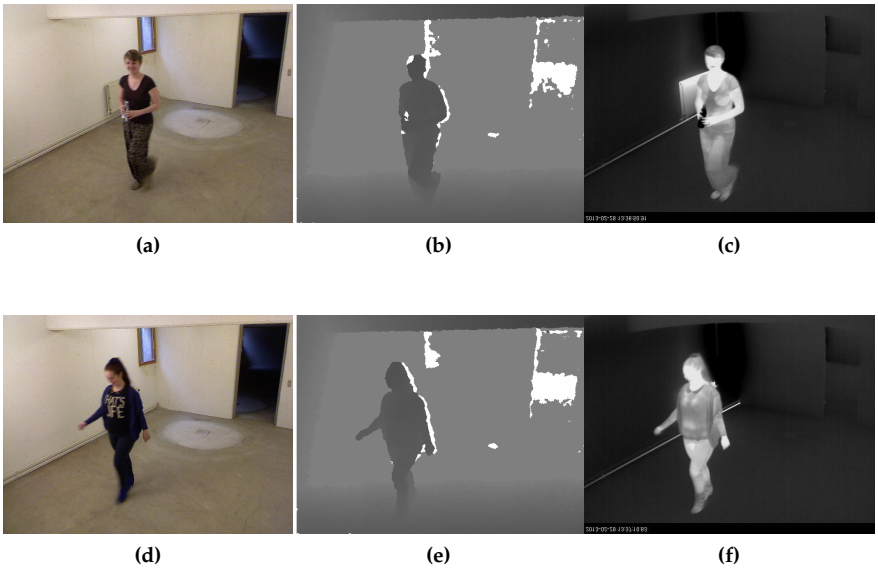


Fig. A.5: Sample images from the tri-modal dataset. left, middle, and right are RGB, depth, and thermal, respectively.

If the user has been determined as already known, it means that the majority of votes are given to a particular user ID which is the one assigned in the re-identification task.

6 Evaluation

Several re-identification datasets with RGB [10, 15] and RGB-D data [2] exist, but to the best of our knowledge no dataset containing all three modalities exists. We have therefore recorded a novel re-identification tri-modal dataset.

The dataset consists of 35 people passing by the sensors twice for 70 passes in total. The vantage point is up and slightly off to the side to mimic a classic surveillance camera setup. All images are 640×480 pixel. Some sample images from each modality are shown in Figure A.5.

The tests were conducted by first extracting the aforementioned features from all passes. As this system is a re-identification system with online enrollment, there is no explicit training phase. Instead, the persons are enrolled if they are very different from previous seen persons.

Since the order of passing will influence the re-identification performance, the system was tested in a random 5-cross validation. We tried the different sets of modalities as input features and found that the best combination of features is the late fusion considering the three sets of modalities with weights:

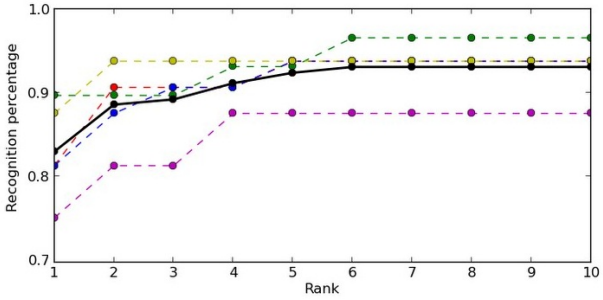


Fig. A.6: CMC-curve performance.

$\alpha = \frac{1}{3}$, $\beta = \frac{1}{3}$, and $\gamma = \frac{1}{3}$ to fit the tri-modal scheme. The results are presented both individually and averaged in terms of: A) passes correctly classified as a new person, B) passes wrongly classified as a new person, C) the number of correctly re-identified persons, D) the number of wrongly re-identified persons, and E) the number of persons ignored, see Table A.1.

If an application requires every single person to be re-identified, then it can be inferred from the table that the performance of our system is 39.4%. In most cases, however, re-identification is used to measure the overall flow and the important issue is therefore to have an acceptable number of true positives and a low number of false positives, where especially the latter is clearly obtained in our system. For comparison a commercial re-identification system based on Wi-Fi signals from smartphones operates with a performance of approximately 50% [4].

	A	B	C	D	E
Run 1	35	10	16	0	9
Run 2	34	12	12	1	11
Run 3	33	13	13	1	10
Run 4	34	12	15	1	8
Run 5	34	10	13	2	11
Average	34	11.4	13.8	1	11
Percentage			93.2%	6.8%	

Table A.1: Re-identification results.

Similar to others working on re-identification we also compute the CMC-curve to show the recognition performance for different rank values, see Figure A.6. Each of the dashed lines is a CMC-curve for a single run. The thick black line is the mean CMC of the 5 runs.

Since this is the first work on tri-modal re-identification we cannot compare our results directly with those of others. Instead in Table A.2 we list the rank-1 results of previous works. Please note that very different datasets and setting were used in these works and that no final conclusions therefore can be drawn.

The results, however, seem to indicate the quality of our tri-modal approach, especially since we do not have a training phase as most others do.

Work	[1]	[2]	[6]	[7]	[8]	[10]	[13]	[20]	[22]	Our
Data	RGB	Depth	RGB	RGB	RGB	RGB	Thermal	RGB-D	RGB	RGB-D-T
Rank-1	51%	12%	N/A	82%	67%	43%	98%	78%	26%	82%

Table A.2: Data types and rank-1 results of recent re-identification works. Note that several works test on a number of different settings and different datasets. In such cases the table contains the average of the best results.

7 Concluding remarks

We proposed a tri-modal re-identification system based on RGB, depth, and thermal descriptors. Three modalities were aligned, and robust discriminative features codifying soft biometrics were computed. The modalities were combined in a late fusion fashion, being able to predict a new user in the scene as well as to recognize previous users based on a combined rule cost. We tested our tri-modal re-identification system on anovel tri-modal dataset. Our results showed that the combination of all three modalities is the one that achieved better performance. A place to improve the system is in the determination of new persons. Nearly all new persons are detected as such, but there is a substantial amount of wrong New Persons. That is not a big issue with regards to re-identification performance, as presumably they will also be difficult to re-identify (they are only detected as new because they are not similar to the known persons), and in many applications it is not critical to be able to re-identify each and every subject. However, fewer wrong New Persons will result in a lower absolute re-identification rate.

References

[1] S. Bak, G. Charpiat, E. Corvée, F. Brémond, and M. Thonnat, "Learning to Match Appearances by Correlations in a Covariance Metric Space," in *ECCV (3)*, ser. LNCS, vol. 7574. Springer, 2012, pp. 806–820. [Online]. Available: http://dblp.uni-trier.de/db/conf/eccv/eccv2012-3.html#BakCCBT12;http://dx.doi.org/10.1007/978-3-642-33712-3_58;http://www.bibsonomy.org/bibtex/202695f9b1a058410faf905a7cf03ba34/dblp

[2] I. B. Barbosa, M. Cristani, A. D. Bue, L. Bazzani, and V. Murino, "Re-identification with RGB-D Sensors." in *ECCV Workshops (1)*, ser. LNCS, vol. 7583. Springer, 2012, pp. 433–442. [Online]. Available: http://dblp.uni-trier.de/db/conf/eccv/eccv2012w1.html#BarbosaCBBM12;http://dx.doi.org/10.1007/978-3-642-33863-2_43;http://www.bibsonomy.org/bibtex/2890ca306be85bd7ed5cdb68a09182f4c/dblp

[3] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314207001555;http://www.bibsonomy.org/bibtex/2dfea172dfaca0272dcdc66acf92ec58d/daill

[4] Blip Systems, "Urban Planning," http://www.bliptrack.com/urban/products/-bliptracktm-sensor/, 2013.

[5] G. Bradski and A. Kaehler, *Learning OpenCV*. O'Reilly, 2008, ch. 7, pp. 201–202.

[6] M. Demirkus, K. Garg, and S. Guler, "Automated person categorization for video surveillance using soft biometrics," pp. 76 670P–76 670P–12, 2010. [Online]. Available: http://dx.doi.org/10.1117/12.851424

[7] G. Doretto, T. Sebastian, P. H. Tu, and J. Rittscher, "Appearance-based person reidentification in camera networks: problem overview and current approaches," *J. Ambient Intelligence and Humanized Computing*, vol. 2, no. 2, pp. 127–151, 2011. [Online]. Available: http://dblp.uni-trier.de/db/journals/jaihc/jaihc2.html#DorettoSTR11;http://dx.doi.org/10.1007/s12652-010-0034-y;http://www.bibsonomy.org/bibtex/2efba953fbb5955b9a686b63ab749e3f8/dblp

[8] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani, "Person re-identification by symmetry-driven accumulation of local features," in *CVPR*, 2010. [Online]. Available: http://dblp.uni-trier.de/db/conf/cvpr/cvpr2010.html#FarenzenaBPMC10;http://dx.doi.org/10.1109/CVPR.2010.5539926;http://www.bibsonomy.org/bibtex/2c2d4e6061cbab23d25067a19ded12eeb/dblp

[9] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge Univ Press, 2000, vol. 2.

[10] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof, "Person Re-identification by Descriptive and Discriminative Classification," in *SCIA*, ser. Lecture Notes in Computer Science, vol. 6688. Springer, 2011, pp. 91–102. [Online]. Available: http://dblp.uni-trier.de/db/conf/scia/scia2011.html#HirzerBRB11;http://dx.doi.org/10.1007/978-3-642-21227-7_9;http://www.bibsonomy.org/bibtex/258f133e3a79dc657b5a918ca1bc42658/dblp

[11] B. Iglewicz and D. Hoaglin, *How to Detect and Handle Outliers*, ser. ASQC basic references in quality control. ASQC Quality Press, 1993. [Online]. Available: http://books.google.es/books?id=siInAQAAIAAJ

[12] K. Jüngling and M. Arens, "Local Feature Based Person Reidentification in Infrared Image Sequences." in *AVSS*. IEEE Computer Society, 2010, pp. 448–455. [Online]. Available: http://dblp.uni-trier.de/db/conf/avss/avss2010.html#JunglingA10;http://doi.ieeecomputersociety.org/10.1109/AVSS.2010.75;http://www.bibsonomy.org/bibtex/209252ac9d464c0cd083747e389e2a660/dblp

[13] ——, "A multi-staged system for efficient visual person reidenticationl," in *Conference on Machine Vision Applications, Nara, Japan*, 2011.

[14] S. J. Krotosky and M. M. Trivedi, "Mutual information based registration of multimodal stereo videos for person tracking," *Computer Vision and Image Understanding*, vol. 106, no. 2, pp. 270–287, 2007.

[15] C. C. Loy, T. Xiang, and S. Gong, "Time-Delayed Correlation Analysis for Multi-Camera Activity Understanding," *Int. J. Comput. Vision*, vol. 90, no. 1, pp. 106–129, Oct. 2010. [Online]. Available: http://dx.doi.org/10.1007/s11263-010-0347-5

[16] B. E. Moore, S. Ali, R. Mehran, and M. Shah, "Visual crowd surveillance through a hydrodynamics lens," *Commun. ACM*, vol. 54, no. 12, pp. 64–73, 2011. [Online]. Available: http://dblp.uni-trier.de/db/journals/cacm/cacm54.html#MooreAMS11;http://doi.acm.org/10.1145/2043174.2043192;http://www.bibsonomy.org/bibtex/2e911f5c0226885d3b9f9949f36c59311/dblp

[17] M. Robnik-Sikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF. Machine Learning," *Machine Learning*, vol. 53, pp. 23–69, 2003.

[18] J. Shotton, A. W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR*. IEEE, 2011, pp. 1297–1304. [Online]. Available: http://dblp.uni-trier.de/db/conf/cvpr/cvpr2011.html#ShottonFCSFMKB11;http://dx.doi.org/10.1109/CVPR.2011.5995316;http://www.bibsonomy.org/bibtex/2d1e61f390b6965f432bdd65ba2ef7c75/dblp

[19] B. Solmaz, B. E. Moore, and M. Shah, "Identifying Behaviors in Crowd Scenes Using Stability Analysis for Dynamical Systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 2064–2070, 2012. [Online]. Available: http://dblp.uni-trier.de/db/journals/pami/pami34.html#SolmazMS12;http://doi.ieeecomputersociety.org/10.1109/TPAMI.2012.123;http://www.bibsonomy.org/bibtex/2aba69affa8b436ee3be376d7824a6fb3/dblp

[20] C. Velardo and J. Dugelay, "Improving Identification by Pruning: A Case Study on Face Recognition and Body Soft Biometric," in *WIAMIS*. IEEE, 2012, pp. 1–4. [Online]. Available: http://dblp.uni-trier.de/db/conf/wiamis/wiamis2012.html#VelardoD12;http://dx.doi.org/10.1109/WIAMIS.2012.6226747;http://www.bibsonomy.org/bibtex/249d2c6092fa27c51ecc485bdc5db7000/dblp

[21] S. Vidas, R. Lakemond, S. Denman, C. Fookes, S. Sridharan, and T. Wark, "A mask-based approach for the geometric calibration of thermal-infrared cameras," *Instrumentation and Measurement, IEEE Transactions on*, vol. 61, no. 6, pp. 1625–1635, 2012.

[22] W. Zheng, S. Gong, and T. Xiang, "Person re-identification by probabilistic relative distance comparison," in *CVPR*. IEEE, 2011, pp. 649–656. [Online]. Available: http://dblp.uni-trier.de/db/conf/cvpr/cvpr2011.html#ZhengGX11;http://dx.doi.org/10.1109/CVPR.2011.5995598;http://www.bibsonomy.org/bibtex/21f384e3d6104d121a22d101f36b43fbe/dblp

Paper B

Multi-modal RGB-Depth-Thermal Human Body Segmentation

Cristina Palmero, Albert Clapés, Chris Bahnsen, Andreas Møgelmose, Thomas B. Moeslund, and Sergio Escalera

The paper has been published in the
International Journal of Computer Vision, Vol. 118, Iss. 2, 2016-6, pp. 217–239, 2016.

Please note that the above article is updated slightly. The updates are:

- An additional reference has been included [19].
- 74 citations to this reference are added.

The additional reference is a Master's thesis: Cristina Palmero Cantariño, *Tri-modal Human Body Segmentation*, Master's thesis, Universitat de Barcelona, Spain, 2014.
There is an overlap between the content of that thesis and the content of the article. The citations are added to make this overlap evident.

The layout has been revised.

Abstract

This work addresses the problem of human body segmentation from multi-modal visual cues as a first stage of automatic human behavior analysis. We propose a novel RGB-Depth-Thermal dataset along with a multi-modal segmentation baseline. The several modalities are registered using a calibration device and a registration algorithm. Our baseline extracts regions of interest using background subtraction, defines a partitioning of the foreground regions into cells, computes a set of image features on those cells using different state-of-the-art feature extractions, and models the distribution of the descriptors per cell using probabilistic models. A supervised learning algorithm then fuses the output likelihoods over cells in a stacked feature vector representation. The baseline, using Gaussian Mixture Models for the probabilistic modeling and Random Forest for the stacked learning, is superior to other state-of-the-art methods, obtaining an overlap above 75% on the novel dataset when compared to the manually annotated ground-truth of human segmentations.

1 Introduction

Human body segmentation is the first step used by most human activity recognition systems [71]. Indeed, an accurate segmentation of the human body and correct person identification are key to successful posture recovery and behavior analysis tasks, and they *benefit the development of a new generation of potential applications in health, leisure, and security* [19].

Despite these advantages, segmentation of people in images poses a challenge to computer vision. The main difficulties arise from the articulated nature of the human body, changes in appearance, lighting conditions, partial occlusions, and the presence of background clutter. Although extensive research has been done on the subject, some constraints must be considered. The researcher must often *make assumptions about the scenario where the segmentation task is to be applied, such as static versus moving camera and indoor versus outdoor location, among other factors. Ideally, it should be tackled in an automatic fashion rather than rely on user intervention, which makes such tasks even more challenging*. [19]

Most state-of-the-art methods that deal with such task *use color images recorded by RGB cameras as the main cue for further analysis, although they present several widely known intrinsic problems, such as similarities in the intensity of background and foreground. More recently, the release of RGB-Depth devices such as Microsoft Kinect® and the new Kinect 2 for Windows® has allowed the community to use RGB images along with per-pixel depth information* [19]. Furthermore, thermal imagery is becoming a complementary and affordable visual modality. Indeed, *having different modalities and descriptions allow us to fuse them to have a more informative and richer representation of the scene. In particular, color modality*

adds contour and texture information and depth data provides the geometry of the scene [19], while thermal imaging adds temperature information.

In this paper we present a novel dataset of RGB-Depth-Thermal video sequences that contains up to three individuals who appear concurrently in three indoor scenarios, performing diverse actions that involve interaction with objects. Sample imagery of the three recorded scenes is depicted in Fig. B.1. The dataset is presented *along with an algorithm that performs the calibration and registration among modalities. In addition, we propose a baseline methodology to automatically segment human subjects appearing in multi-modal video sequences* [19]. We start reducing the search space by learning a model of the scene to subsequently perform background subtraction, thus segmenting subject candidate regions in all available and registered modalities. Such regions are then described using simple but reliable uni-modal feature descriptors. *These descriptors are used to learn probabilistic models so as to predict the* [19] candidate region that actually belongs to people. In particular, likelihoods obtained from a set of Gaussian Mixture Models (GMMs) are fused in a higher level representation and modeled using a Random Forest classifier. *We compare results from applying segmentation to the different modalities separately to results obtained by fusing features from all modalities* [19]. In our experiments, we demonstrate the effectiveness of the proposed algorithms to perform registration among modalities and to segment human subjects. *To the best of our knowledge, this is the first publicly available dataset and work that combines color, depth, and thermal modalities to perform the people segmentation task in videos, aiming to bring further benefits towards developing new – and more robust – solutions* [19].

The remainder of this paper is organized as follows: Section 2 reviews the different approaches for human body segmentation that appear in the recent literature [19]. Section 3 presents the new dataset, including acquisition details, the calibration device, the registration algorithm, and the ground-truth annotation. Section 4 presents the proposed baseline methodology for multi-modal human body segmentation, which is experimentally evaluated in Section 5 along with the registration algorithm. We present our conclusions in Section 6.

2 Related work

Multi-modal fusion strategies have gained attention lately due to the decreasing price of sensors. They are usually based on existing modality-specific methods that, once combined, enrich the representation of the scene in such a way that the weaknesses of one modality are offset by the strengths of another. Such strategies have been successfully applied to the human body segmentation task, *which is one of the most widely studied problems in computer vision* [19].

Fig. B.1: Three views of each of the three scenes shown in the RGB, thermal, and depth modalities, respectively.

In this section we focus on the most recent and relevant studies, techniques and methods of individual and multi-modal human body segmentation. We also review the existing multi-modal datasets devoted to such task.

Color methods. Background subtraction is one of the most applied techniques when dealing with image segmentation in videos. The parametric model that [83] proposed, *which models the background using a mixture of gaussians (MoG), has been widely used, and many variations based on it have been suggested. [10] thoroughly reviewed more advanced statistical background modeling techniques. Nonetheless, after obtaining the moving object contours one still needs a way to assess whether they belong to a human entity. Human detection methods are strongly related to the task of human body segmentation because they allow us to discriminate better among other objects. They usually produce a bounding box that indicates where the person is, which in turn may be useful as a prior for pixel-based or bottom-up approaches to refine the final human body silhouette. In the category of holistic body detectors, one of the most successful representations is the Histogram of Oriented Gradients (HOG) [26], which is the basis of many current detectors. Used along with a discriminative classifier – e.g. Support Vector Machines (SVM) – it is able to accurately predict the presence of human subjects. Example-based methods [5] have also been proposed to address human detection, utilizing templates to compare*

the incoming image and locate the person but limiting the pose variability [19].

In terms of descriptors, other possible representations, apart from the already commented HOG, are those that try to fit the human body into silhouettes [60], those that model color or texture such as Haar-like wavelets [90], optical flow quantized in Histograms of Optical Flow (HOF) [27], and, more recently, descriptors including logical relations, e.g. Grouplets [99], which enable observers to recognize human-object interactions [19].

Instead of whole body detection, some approaches have been built on the assumption that the human body consists of an ensemble of body parts [69,74]. Some of these are based on pictorial structures [4,97]. In particular, [97], [98], and [33] outperform other existing methods using a Deformable Part-based Model (DPM). This model consists of a root HOG-like filter and different part-filters that define a score map of an object hypothesis, using latent SVM as a classifier. Another well-known part-based detector is Poselets [9,93], which trains different homonymous parts to fire at a given part of the object at a given pose and viewpoint. More recently, [91] have proposed Motionlets for human motion recognition. Grammar models [40] and AND-OR graphs [102] have been also used in this context [19].

Other approaches model objects as an ensemble of local features. This category includes methods such as Implicit Shape Models (ISM) [52], which consist of visual words combined with location information. These are also used in works that estimate the class-specific segmentation based on the detection result after a training stage [53] [19].

Conversely, generative classifiers deal directly with the person segmentation problem. They function in a bottom-up manner, learning a model from an initial prior in the form of bounding boxes or seeds, and using it to yield an estimate for the background and target distributions, normally applying Expectation Maximization (EM) [20, 80]. One of the most popular is GrabCut [42, 75], an interactive segmentation method based on Graph Cuts [12] and Conditional Random Fields (CRF) that combines pixel appearance information with neighborhood relations to refine silhouettes, using a bounding box as an initialization region [19].

Having considered the properties of each of the aforementioned segmentation categories, it is understandable that a combination of several approaches would be proposed, namely top-down and bottom-up segmentation [36,51,54,57,63]. To name just a few, ObjCut [50] combines pictorial structures and Markov Random Fields (MRF) to obtain the final segmentation. PoseCut [14] is also based on MRF and Graph Cuts but has the added ability to deal with 3D pose estimation from multiple views [19].

Depth methods. Most of the aforementioned contributions use RGB as the principal cue to extract the corresponding descriptors. *The recent release of affordable RGB-Depth devices such as Microsoft® Kinect®* [19] has encouraged the community to start using depth maps as a new source of information. [81] was one of the first contributions, which used depth images to extract the human body pose, an approach that is also the core of the Kinect® human

recognition framework.

A number of standard computer vision methods already mentioned for color cues have been applied to depth maps. For example, a combination of Graph Cuts and Random Forest has been applied to part-based human segmentation [44]. [46] proposed the use of *Poselets* as a representation that combines part-based and example-based estimation aspects for human pose estimation. Generative models have also been considered, such as in [21], where they are used to learn limb shape models from depth, silhouette and 3D pose data. Active Shape Models (ASM), Gabor filters [73], template matching, geodesic distances [77], and linear programming [94] have also been employed in this context.

Notwithstanding the former, the emergence of the depth modality has lead to the design of novel descriptors. [70], for example, proposed a key-point detector based on the saliency of depth maps for identifying body parts. Point feature histograms, based on the orientations of surface normal vectors and taking advantage of a 3D point cloud representation, have also been proposed for local body shapes representation [43]. [96] applied a 2D Chamfer match over silhouettes for human detection and segmentation based on contouring depth images. A more recent contribution is the Histogram of Oriented 4D Normals (HON4D) [67], which proposes a histogram that captures the distribution of the surface normal orientations in the 4D space of depth, time, and spatial coordinates. Recently, [58] presented a method that describes hand poses by a 3D spherical descriptor of cloud density distributions.

Thermal methods. *In contrast to color or depth cues, thermal infrared imagery has not been used widely for segmentation purposes, although it is attracting growing interest by the research community. Several specific descriptors have been proposed. For example, HOG and SVM are used in [85]* [19], while [100] extended such combination with *Edgelets* and AdaBoost. Other examples include joint shape and appearance cues [25], probabilistic models [7], Shape Context Descriptor (SCD) with AdaBoost [92], and descriptors invariant to scale, brightness and contrast [66]. *Background subtraction has also been adapted to deal with this kind of imagery [28]. In that study, the authors presented a statistical contour-based technique that eliminates typical halo artifacts produced by infrared sensors by combining foreground and background gradient information into a contour saliency map in order to find the strongest salient contours* [19]. An example of human segmentation is found in [34], which applies thresholding and shape analysis methods to perform such task.

Most of the cited contributions focus on pedestrian detection applications. Indeed, thermal imaging has attracted the most attention for occupancy analysis [37] and pedestrian detection applications, due to the cameras' ability to see without visible illumination and the fact that people cannot be identified in thermal images, which eliminates privacy issues. In addition to these, a key advantage of thermal imaging for detecting people is its discriminative power,

due to the big difference in heat intensity where a human is present.

For more, we refer the reader to [38], an extensive survey of thermal cameras and more applications, including technological aspects and the nature of thermal radiation [19].

Combining modalities. Given the increasing popularity of depth imagery, it is not surprising that a number of algorithms that combine both depth and RGB cues have appeared to benefit from multi-modal data representation [2, 23, 43, 76, 78, 79, 84, 87]. A recent example is *PoseField* [89], a filter-based mean-field inference method that jointly estimates human segmentation poses, per-pixel body parts, and depth, given stereo pairs of images. Indeed, disparity computation from stereo images is another widely-used approach for obtaining depth maps without range and outdoor limitations. Even background subtraction approaches can profit from such a fusion, since it is possible to reduce those misdetections that cannot be tackled by each modality individually [18, 35, 39, 41].

Similar to the RGB-Depth combination, thermal imaging has also been fused with color cues to enrich data representation. Such combinations have been applied to pedestrian tracking [55, 56], in which the authors apply a codeword-based background subtraction model and a Kalman filter to track pedestrian candidates. The pedestrian classification is handled by a symmetry analysis based on a Double Helical Signature. In [29], Contour Saliency Maps are used to improve a single-Gaussian background subtraction. RGB-Thermal human body segmentation is tackled by [101] and, unlike the previously described approaches, the authors' dataset contains objects in close range of the cameras. This means that one cannot rely on a fixed transformation to register the modalities. Instead, the geometric registration is performed at a blob level between visual objects corresponding to human subjects.

Only a few scholars have considered the fusion of RGB, depth, and thermal features (RGB-D-T) to improve detection and classification capabilities. The latest contributions include people following, human tracking, re-identification, and face recognition. [86] used a laser scanner, along with the RGB-D-T sensors, for people detection and people following. The detection is performed separately on each modality and fused on a decision level. [22] performed RGB-D-T human motion tracking to determine the 2D position and orientation of people in a constrained, indoor scenario. In [62], features extracted on the three modalities are combined to perform person re-identification. More recently, [65] performed RGB-D-T face recognition based on Local Binary Patterns, HOG, and HAAR-features. [48] provide an interesting approach by using spatiotemporal features and combining the three modalities to estimate pain level from facial images. However, little attention has been paid to human segmentation applications combining such cues.

Existing datasets. Up to this point we have extensively reviewed methods related to multi-modal human body segmentation. Such task is often a first

step towards further sophisticated pose and behavior analysis approaches. *To advance research in this area, it is necessary to have the right means to compare methods so as to measure improvements. There are several static and continuous image-based human-labeled datasets that can be used for that purpose [61], which try to provide realistic settings and environmental conditions. The best known of these is the Berkeley Segmentation Dataset and Benchmark [59], which consists of 12,000 segmented items of 1,000 Corel dataset color images containing people or different objects. It also includes figure-ground labelings for a subset of the images. [3] also made available a database containing 200 gray level images along with ground-truth segmentations. This dataset was specially designed to avoid potential ambiguities by incorporating only those images that clearly depict one or two objects in the foreground that differ from their surroundings in terms of texture, intensity, or other low level cues. However, the dataset does not represent uncontrolled scenarios. The well known PASCAL Visual Object Classes Challenge [30] tended to include a subset of the color images annotated in a pixel-wise fashion for the segmentation competition. Although not considered to be benchmarks, Kinect-based datasets are also available, and this device is widely used in human pose related works. [42] presented a novel dataset consisting of 3,386 images of segmented humans and ground-truth automatically created by Kinect®, which consists of different human subjects across four different locations. Unfortunately, depth map images are not included in the public dataset* [19].

Despite this large body of work, little attention has been given to multi-modal video datasets. *We underline the collective datasets of Project ETISEO [64], owing to the fact that for some of the scenes the authors include an additional imaging modality, such as infrared footage, in addition to color images. It consists of indoor and outdoor scenes of public places such as an airport apron or a subway station, as well as a frame-based annotated ground-truth. Depth maps computed from stereo pairs of images are used in INRIA 3D Movie dataset [2], which contains sequences from 3D movies. Such sequences show people performing a broad variety of activities from a range of orientations and with different levels of occlusions* [19]. A comparison of existing multi-modal datasets focused on human body related approaches is provided in Table B.1. As one can see, there is a lack of datasets that combine RGB, depth, and thermal modalities focused on the human body segmentation task, like the one we propose in this paper.

3 The RGB-Depth-Thermal dataset

The proposed dataset features a total of 11,537 frames divided into three indoor scenes, of which 5,724 are annotated. Having pictured sample imagery of the three scenes in Fig. B.1, we also show their corresponding number of annotated frames and depth range in Table B.2. Activity in scene 1 and 3 uses the full depth range of the Kinect® sensor, whereas activity in scene 2 is constrained to a depth range of ± 0.250 meters in order to suppress the

Dataset	Data Format	Video Seq.	Annotation	Scenario	Purpose
ETISEO Project [64]	RGB-T	Yes	Bounding Box	Indoor/ Outdoor	Video Surveillance
IRIS Thermal/Visible Face Database [1]	RGB-T	No	-	Indoor	Face Detection
OSU Color-Thermal Database [29]	RGB-T	Yes	Bounding box	Outdoor	Object Detection
RGB-D People dataset [82]	RGB-D	Yes	Bounding Box	Indoor	Human Detection
H2View dataset [79]	RGB-D (stereo)	Yes	Segmentation masks, Ground-truth depth, Human pose	Indoor	3D Pose Estimation
LIRIS Human activities dataset [95]	RGB-D	Yes	Bounding box, Activity class	Indoor	Human Activity Recognition
RGB-D Person Re-identification dataset [6]	RGB-D	Yes	Foreground masks, Skeleton, 3D mesh	Indoor	Person Re-identification
VAP RGB-D Face dataset [45]	RGB-D	No	Pose class	Indoor	Face Detection, Pose Estimation
Biwi Kinect Head Pose Database [31]	RGB-D	Yes	Head 3D position, Head rotation	Indoor	Head Pose Estimation
Cornell Activity datasets [49]	RGB-D	Yes	Bounding box, Activity class, Skeleton	Indoor	Human Activity Recognition
Eurecom Kinect Face dataset [47]	RGB-D	No	6 facial landmarks, Person information	Indoor	Face Recognition
Inria 3D Movie dataset [2]	RGB-D (stereo)	Yes	Bounding box, Human pose, Segmentation masks	Indoor/ Outdoor	Human Detection, Human Segmentation, Pose Estimation
RGB-D-T Facial Database [65]	RGB-D-T	No	Bounding box	Indoor	Face Recognition
Our proposal	RGB-D-T	Yes	Pixel-level	Indoor	Human detection, Human Segmentation, Person Re-identification

Table B.1: Comparison of multi-modal datasets aimed for human body related approaches in order of release.

parallax between the two physical sensors. Scenes 1 and 2 are situated in a closed meeting room with little natural light to disturb the sense of depth, while scene 3 is situated in an area with wide windows and a substantial amount of sunlight. The human subjects are walking, reading, using their phones, and, in some cases, interacting with each other. In all scenes, at least one of the humans interacts with a heated object in order to complicate the extraction of humans in the thermal domain. Examples of heated objects in the scene are radiator pipes, boilers, toasters, and mugs.

Scene	Frames	Annotated frames	Depth range
1	4693	1767	1-4 m
2	2216	2016	1.4-1.9 m
3	4628	1941	1-4 m

Table B.2: Annotated number of frames and spatial constraints of the scenes in meters (m).

3.1 Acquisition

The RGB-D-T data stream is recorded using a Microsoft® Kinect® for XBOX360, which captures the RGB and depth image streams, and an AXIS Q1922 thermal camera. The resolution of the imagery is fixed at 640 $\times$ 480 pixels. As seen in Fig. B.2, the cameras are vertically aligned in order to reduce perspective distortion.

Fig. B.2: Camera configuration. The RGB and thermal sensor are vertically aligned.

The image streams are captured using custom recording software that invokes the Kinect for Windows® and AXIS Media Control SDKs. The integration of the two SDKs enables the cameras to be calibrated against the same system clock, which enables the post-capture temporal alignment of the image streams. Both cameras are able to record at 30 FPS. However, the dataset is recorded at 15 FPS due to recording software performance constraints.

3.2 Multi-modal calibration

The calibration of the thermal and RGB cameras was accomplished using a thermal-visible calibration device inspired by [88]. The calibration device consists of two parts: we use an A3-sized 10 mm polystyrene foam board as a backdrop and a board of the same size with cut-out squares as the checkerboard. Before using the calibration device, we heat the backdrop and keep the checkerboard plate at room temperature, thus maintaining a suitable thermal contrast when joined, as seen in Fig. B.3. Using the Camera Calibration Toolbox of [8], we are able to extract corresponding points in the thermal and RGB modalities. The sets of corresponding points are used to undistort both image streams and for the subsequent registration of the modalities.

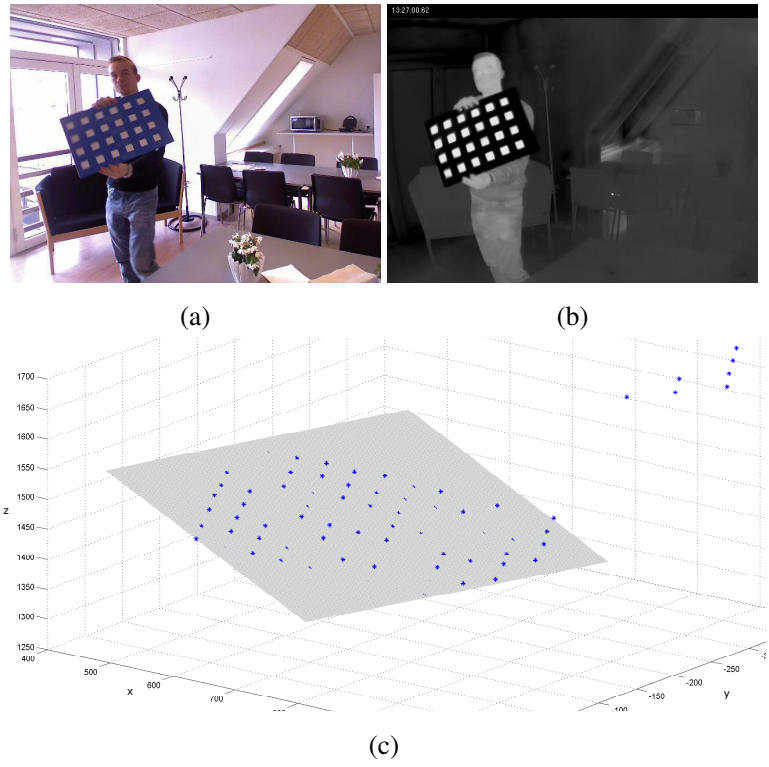


Fig. B.3: The calibration device as seen by the (a) RGB and (b) thermal camera. The corresponding points in world coordinates and the plane, which induces a homography, are overlayed in (c). Noise in the depth information accounts for the outliers in (c).

3.3 Registration

The depth sensor of the Kinect® is factory registered to the RGB camera and a point-to-point correspondence is obtained from the SDK. The registration is static and might therefore be saved in two look-up-tables for RGB ⇔ depth.

The registration from RGB ⇒ thermal, $\mathbf{x} \Rightarrow \mathbf{x}'$, is handled using a weighted set of multiple homographies based on the approximate distance to the view that the homography represents. By using multiple homographies, we can compensate for parallax at different depths. However, the spatial dependency of the registration implies that no fixed, global registration or look-up-table is possible, thus inducing a unique mapping for each pixel at different depths.

Homographies relating RGB and thermal modalities are generated from a minimum of 50 views of the calibration device scattered throughout each scene. One view of the calibration device induces 96 sets of corresponding points in the RGB and thermal modality (Fig. B.3c), from which a homography is computed using a RANSAC-based method. The acquired homography and the registration it establishes are only accurate for points on the plane that are spanned by the particular view of the calibration device. To register an arbitrary point of the scene, $\mathbf{x} \Rightarrow \mathbf{x}'$, the 8 closest homographies are weighted according to this scheme:

1. For all J views of the calibration device, calculate the 3D centre of the K extracted points in the image plane:

$$\overline{\mathbf{X}}_j = \frac{\sum_{k=1}^{K} \mathbf{X}_{k_j}}{K} = \frac{\sum_{k=1}^{K} \mathbf{P}^{+}\mathbf{x}_{k_j}}{K}. \tag{B.1}$$

   The depth coordinate of $\mathbf{X}$ is estimated from the registered point in the depth image. $\mathbf{P}^{+}$ is the pseudoinverse of the projection matrix.

2. Find the distance from the reprojected point $\mathbf{X}$ to the homography centres:

$$\omega(j) = |\mathbf{X} - \overline{\mathbf{X}}_j|. \tag{B.2}$$

3. Centre a 3D coordinate system around the reprojected point $\mathbf{X}$ and find $\min \omega(j)$ for each octant of the coordinate system. Set $\omega(j) = 0$ for all other weights. Normalize the weights:

$$\omega^*(j) = \frac{\omega(j)}{\sum_{j=1}^{J} \omega(j)}. \tag{B.3}$$

4. Perform the registration $\mathbf{x} \Rightarrow \mathbf{x}'$ by using a weighted sum of the homographies:

$$\mathbf{x}' = \sum_{j=1}^{J} \omega^*(j)\, \mathbf{H}_j \mathbf{x}, \tag{B.4}$$

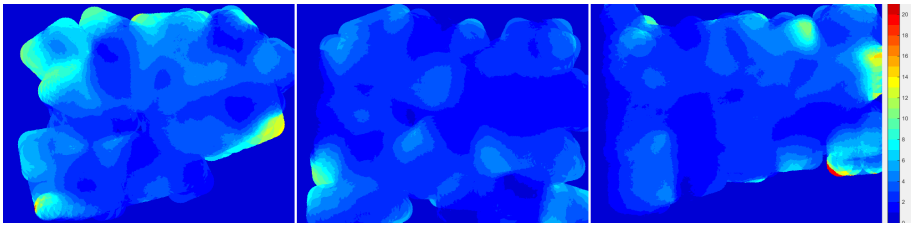


Fig. B.4: Average registration error, RGB (a) $\Rightarrow$ thermal (b), of the three dataset sequences, averaged over the depth range of the Kinect. The errors are shown in image coordinates and are computed from multiple views of the calibration device. Registrations errors are more prominent in the boundaries of the images.

where $\mathbf{H}_j$ is the homography induced by the j^{th} view of the calibration device.

For registering thermal points, the absence of depth information means that points are reprojected at a fixed distance, inducing parallax for points at different depths. Thus, the registration framework may be written:

$$\text{depth} \Leftrightarrow \text{RGB} \Rightarrow \text{thermal} \tag{B.5}$$

The accuracy of the registration of RGB $\Rightarrow$ thermal is mainly dependent on:

1. The distance in space to the nearest homography.
2. The synchronization of thermal and RGB cameras. At 15 FPS, the maximal theoretical temporal misalignment between frames is thus 34 ms.
3. The accuracy of the depth estimate.

A quantitative view of the registration accuracy is provided in Fig. B.4. An example of the registration for Scene 3 is seen in Fig. B.5.

3.4 Annotation

The acquired videos were manually annotated frame by frame in a custom annotation program called Pixel Annotator. The dataset contains a large number of frames spread over a number of different sequences. All sequences have three modalities: RGB, depth, and thermal. The focus of the annotation is on the people in the scene and a mask-based annotation philosophy was employed. This means that each person is covered by a mask and each mask (person) has a unique ID that is consistent over all frames. In this way the dataset can be used not only for subject segmentation, but also for

Fig. B.5: Example of RGB (a) ⇒ thermal (b) registration.

tracking and re-identification purposes. Since the main purpose of the dataset is segmentation, it was necessary to use a pixel-level annotation scheme. Examples of the annotation and registered annotated masks are shown in Fig. B.7.

Pixel Annotator provides a view of each modality with the current mask overlaid, as well as a raw view of the mask (see Fig. B.6). It implements the registration algorithm described above so that the annotator can judge whether the mask fits in all modalities. Because the modalities are registered to each other, there are not specific masks for any given modality but rather a single mask for all.

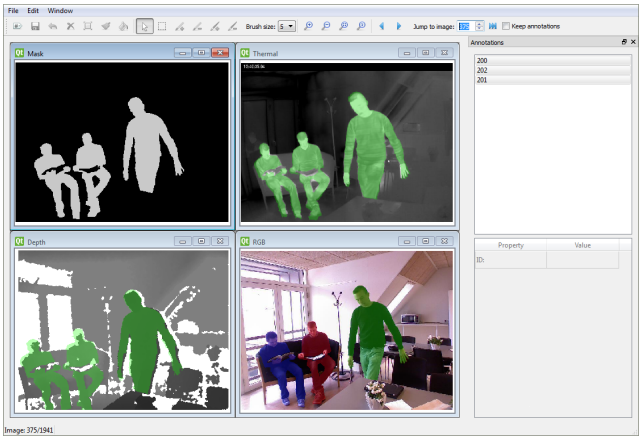


Fig. B.6: Pixel Annotator showing the RGB masks and the corresponding, registered masks in the other views.

Each annotation can be initialized with an automatic segmentation using

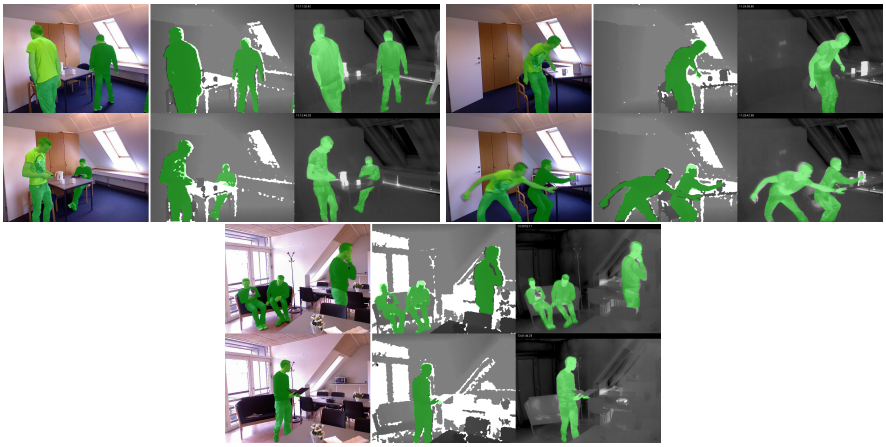


Fig. B.7: Examples of the annotated imagery for two views in each of the three scenes. The RGB modality is manually annotated and the corresponding mask is registered to the depth and thermal modalities. The causes of registration misalignment of the masks are motion blur and noisy depth information, which induce parallax in the thermal modality.

the GrabCut algorithm [75] to get it quickly off the ground. Pixel Annotator then provides pixel-wise editing functions to further refine the mask. Each annotation is associated with a numerical ID and can have an arbitrary number of property fields associated with it. They can be boolean or contain strings so that advanced annotation can take place, from simple occluded/not occluded fields to fields describing the current activity. Pixel Annotator is written in C++ on the Qt framework and is fully cross-platform compatible.

The dataset and the registration algorithm is freely available at `http://www.vap.aau.dk/`. Since we subdivided the several scenes into 10 variable-length sequences in order to carry out our baseline experiments, we also provide the partitionings in a file along with the dataset. We refer the reader to Section 5 for more details about the evaluation of the baseline.

4 Multi-modal human body segmentation

We propose a baseline methodology to segment human subjects automatically in multi-modal video sequences. The first step of our method focuses on reducing the spatial search space by estimating the scene background to extract the foreground regions of interest in each one of the modalities. Note that such regions may belong to human or non-human entities, so in order to perform an accurate classification we describe them using modality-specific state-of-the-art feature descriptors. *The obtained features are then used to learn probabilistic models in order to predict which foreground regions actually belong to*

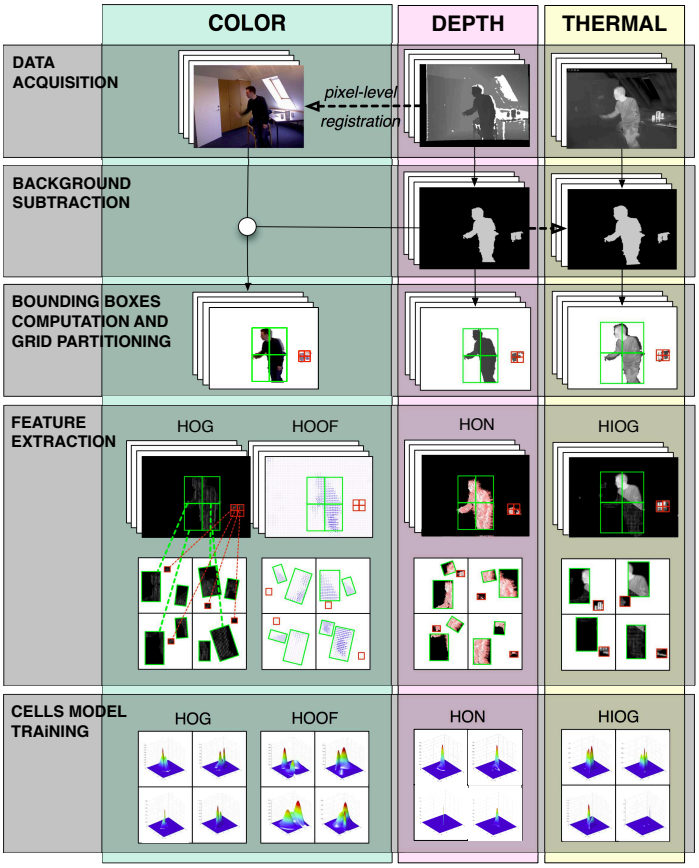


Fig. B.8: The main steps of the proposed baseline method, before reaching the fusion step. Adaption of [19].

human subjects [19]. Predictions obtained from the different models are then fused using a learning-based approach. Fig. B.8 depicts the different stages of the method.

4.1 Extraction of masks and regions of interest

The first step of our baseline is to reduce the search space [19]. For this task, we learn a model of the background and perform background subtraction.

Background subtraction

A widely used approach for background modeling in this context is Gaussian Mixture Models (GMM), which assigns a mixture of gaussians per pixel with a fixed number

of components [11]. Sometimes the background presents periodically moving parts such as noise or sudden and gradual illumination changes. Such problems are often tackled with adaptive algorithms that keep learning the pixel's intensity distribution after the learning stage with a decreased learning rate. However, this also causes intruding objects that stand still for a period of time to vanish, so a non-adaptive approach is more convenient in our case [19].

Although this background subtraction technique performs fairly well, it has to deal with the intrinsic problems of the different image modalities. For instance, color-based algorithms may fail due to shadows, similarities in color between foreground and background, highlighted regions, and sudden lighting changes. Thermal imagery may also have this kind of problems, in addition to the inconvenience of temperature changes in objects. A halo effect can also be observed around warm items. Regarding depth-based approaches, they may produce misdetections due to the presence of foreground objects at a depth similar to that of the background. Depth data is quite noisy and many pixels in the image may have no depth due to multiple reflections, transparent objects, or scattering in certain surfaces such as human tissue and hair. Furthermore, a halo effect around humans or objects is usually perceived due to parallax issues caused by the separation of the infrared emitter and sensor of the Kinect® device. However, they are more robust when it comes to lighting artifacts and shadows. A comparison is shown in Fig. B.9, where the actual foreground objects are the humans and the objects on the table. As one can see, RGB fails at extracting the human legs because they are of a similar color to the chair in the back. The thermal cue segments the human body more accurately, but it includes some undesired reflections and illuminates the jar and mugs with a surrounding halo. The pipe tube is also extracted as foreground due to temperature changes over time [19].

Despite its drawbacks, depth-based background subtraction is the one that seems to give the most accurate result. Therefore, the binary foreground masks of our proposed baseline are computed applying background subtraction to the depth modality previously registered to the RGB one, thereby allowing us to use the same masks for both modalities. Let us consider the depth value of a pixel at frame i as $z^{(i)}$. The background model $p(z^{(i)}|B)$ – where B represents the background – is estimated from a training set of depth images represented by $\mathcal{Z}$ using the T first frames of a sequence such that $\mathcal{Z} = \{z_1^{(i)}, \ldots, z_T^{(i)}\}$. This way, the estimated model is denoted by $\hat{p}(z^{(i)}|\mathcal{Z}, B)$, modeled as a mixture of gaussians. We use the method presented in [103], which uses an on-line clustering algorithm that constantly adapts the number of components of the mixture for each pixel during the learning stage [19].

Extraction of regions of interest

Once the binary foreground masks are obtained, a 2D connected component analysis is performed using basic mathematical morphological operators. We also set a minimum value for each connected component area – except in left and rightmost sides of the image, which may be caused by a new incoming item – to clean the noisy output

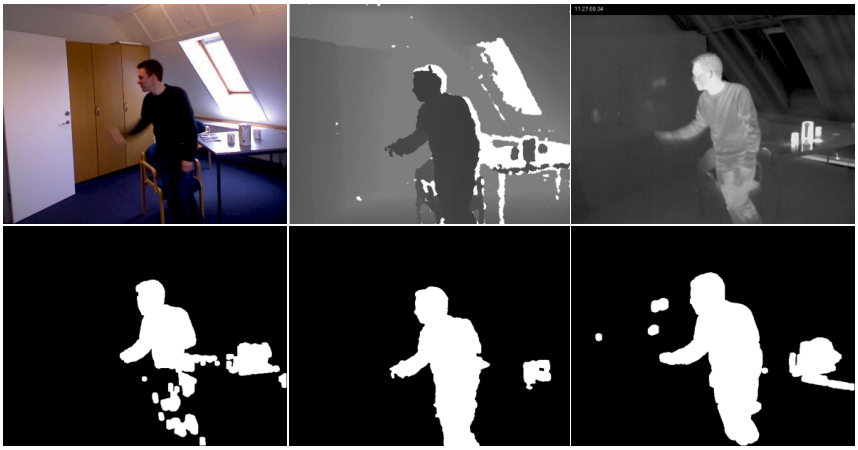


Fig. B.9: Background subtraction for different visual modalities of the same scene (RGB, depth, and thermal respectively). First appeared in [19].

mask [19].

A region of interest should contain a separated person or object. However, different subjects or objects may overlap in space, resulting in a bigger component that contains more than one item. For this reason, each component has to be analyzed to find each item separately in order to obtain the correct bounding boxes that surround them [19].

One of the advantages of the depth cue is that we can use the depth value in each pixel to know whether an item is farther than another. We can assume that a given connected component denotes just one item if there is no rapid change in the disparity distribution and it has a low standard deviation. For those components that do have a greater standard deviation, and assuming a bimodal distribution – two items in that connected component –, Otsu's method [68] can be used to split the blob in two classes such that their intra-class variance is minimal [19].

For such purposes, we define $\mathbf{c}$ *as a vector containing the depth range values that correspond to a given connected component, with mean* $\mu_{\mathbf{c}}$ *and standard deviation* $\sigma_{\mathbf{c}}$*, and* σ_{otsu} *as a parameter that defines the maximum* $\sigma_{\mathbf{c}}$ *allowed to not apply Otsu. Note that erroneous or out-of-range pixels do not have to be taken into account in* $\mathbf{c}$ *when finding the Otsu's threshold because they would change the disparity distribution, thus leading to incorrect divisions. Hence, if* $\sigma_{\mathbf{c}} > \sigma_{\text{otsu}}$*, Otsu is applied. However, the assumption of bimodal distribution may not hold, so to take into account the possibility of more than two overlapping items the process is applied recursively to the divided regions in order to extract all of them* [19].

Once the different items are found, the regions belonging to them are labeled using a different ID per item. In addition, rectangular bounding boxes are generated encapsulating such items individually over time, whose function is to denote the

regions of interest of a given foreground mask [19].

Correspondence to other modalities

As stated in Section 4.1, depth and color cues use the same foreground masks, so we can take advantage of the same bounding boxes for both modalities. Foreground masks for the thermal modality are computed using the provided registration algorithm with the depth/color foreground masks as input [19]. For each frame, each item is registered individually to the thermal modality and then merged into one mask, thus preserving the same item ID for the depth/color foreground masks. In this way, we achieve a one-to-one straightforward correspondence between items of all modalities, and the constraint of having the same number of items in all the modalities is fulfilled. Bounding boxes are generated in the same way depth modality is, which, although they do not have the *same coordinates, denote the same regions of interest. Henceforth, we use R to refer to such regions and* $F = \{F^{\text{color}}, F^{\text{depth}}, F^{\text{thermal}}\}$ *to refer to the set of foreground masks* [19].

Tagging regions of interest

The extracted regions of interest are further analyzed to decide whether they belong to objects or subjects. In order to train and test the models and determine final accuracy results, we need to have a ground-truth labeling of the bounding boxes in addition to the ground-truth masks.

This labeling is done in a semiautomatic manner. First, we extract bounding boxes from regions of interest of ground-truth masks, compare them to those extracted previously from the foreground masks F, and compute the overlap between them. Defining y_r as the label applied to the r region of interest, the automatic labeling is therefore applied as follows:

$$y_r = \begin{cases} 0 & \text{(Object)} & \text{if overlap} \leq \lambda_1 \\ -1 & \text{(Unknown)} & \text{if } \lambda_1 < \text{overlap} < \lambda_2 \\ 1 & \text{(Subject)} & \text{if overlap} \geq \lambda_2 \end{cases} \tag{B.6}$$

In this way, regions with low overlap are considered to be objects, whereas those with high overlap are classified as subjects. A special category named *unknown* has been added to denote those regions that do not lend themselves to direct classification, such as regions with subjects holding objects, multiple overlapping subjects, and so on.

However, such conditions may not always hold, since some regions whose overlap value is lower than λ_1 compared to the ground-truth masks could actually be part of human beings. For this reason we reviewed the applied labels manually to check for possible mislabelling.

4.2 Grid partitioning

Given the accuracy of the registration, particularly because of the depth-to-thermal transformation, we are not able to make an exact pixel-to-pixel correspondence. Instead, the association is made among greater information units: grid cells. In the context of this work, a grid cell is the unit of information processed in the feature extraction and classification procedures [19].

Each region of interest $r \in R$ associated with either a segmented subject or object is partitioned in a grid of $n \times m$ cells. Let G_r denote a grid, which in turn is a set of cells, corresponding to the region of interest r. Hence, we write G_{rij} to refer to the position (i, j) in the r-th region, such that $i \in \{1, ..., n\}$ and $j \in \{1, ..., m\}$ [19].

Furthermore, a grid cell G_{rij} consists of a set of multi-channel images $\{\mathbf{G}_{rij}^{(c)} \mid \forall c \in \mathcal{C}\}$, corresponding to the set of cues [19]:

$$\mathcal{C} = \{\text{“color”}, \text{“motion”}, \text{“depth”}, \text{“thermal”}\} \tag{B.7}$$

Accordingly, $\{\mathbf{G}_{rij}^{(c)} \mid \forall r \in R\}$, i.e. the set of (i, j)-cells in the c cue, is indicated by $G_{ij}^{(c)}$ [19].

The next section provides the details about the feature extraction processes on the different visual modalities at cell level [19].

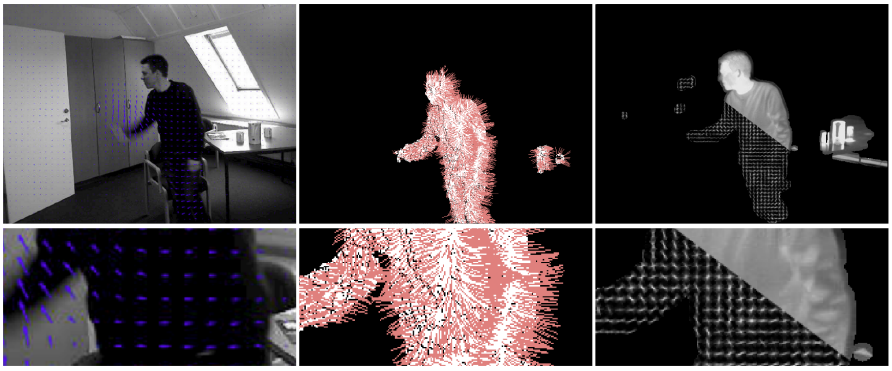
4.3 Feature extraction

Each cue in $\mathcal{C}$ involves its own specific feature extraction/ description processes. For this purpose, we define the feature extraction function f such that $f\colon \mathbb{R}^{n\times m} \rightarrow \mathbb{R}^{\delta}$. Accordingly, $\mathbf{G} \xrightarrow{\mathbb{R}^{n\times m}} \mathbf{d}$, where $\mathbf{d}$ is a δ-dimensional vector, representing the description of $\mathbf{G}$ in a certain feature space (the output space of f). For the color modality two kinds of descriptions are extracted for each cell – Histogram of Oriented Gradients (HOG) and Histogram of Optical Flows (HOF) –, whereas in the depth and thermal modality the Histogram of Oriented Normals (HON) and Histogram of Intensities and Oriented Gradients (HIOG) are used respectively [19]. Hence, we define a set of four different kinds of descriptions $\mathcal{D} = \{\text{HOG}, \text{HOF}, \text{HON}, \text{HIOG}\}$. In this way, for a particular cell G_{rij}, we extract the set of descriptions $D_{rij} = \{f_d(\mathbf{G}_{rij}^{(c)}) \mid c = \varpi(d)\ , \forall d \in \mathcal{D}\} = \{\mathbf{d}_{rij}^{(d)} \mid \forall d \in \mathcal{D}\}$. The function $\varpi(\cdot)$ simply returns the cue corresponding to a given description.

Color modality

The color imagery is the most popular modality and has been extensively used to extract a range of different feature descriptions [19].

Histogram of oriented gradients (HOG). For the color cue, we make the most of the original implementation of HOG but with a lower descriptor



(a) Optical flow (b) Depth normals (c) Thermal intensities (upper right) & oriented gradients (lower left)

Fig. B.10: Example of descriptors computed in a frame for the different modalities: (a) represents the motion vectors using a forward scheme; that is, the optical flow orientation gives insight into where the person is going in the next frame; (b) the computed surface normal vectors; and (c) the thermal intensities and thermal gradients' orientations. Adapted from [19].

dimension than the original by not overlapping the HOG blocks. For the gradient computations, *we use RGB color space with no gamma correction and the Sobel kernel* [19].

The gradient orientation is therefore determined for each pixel by considering the pixel's dominant channel and quantized in a histogram over each HOG-cell (note that we are not referring to our cells), evenly spacing orientation values in the range $[0°, 180°]$. HOG-cells' histograms in each HOG-block are concatenated and L2-normalized. Finally, normalized HOG-block histograms are concatenated in the κ-bin histogram that we use for our cell classification.

Histogram of Optical Flow (HOF). *The color cue also allows us to obtain motion information by computing the dense optical flow and describing the distribution of the resultant vectors. The optical-flow vectors of the whole image can be computed using the luminance information of image pairs with the Gunnar Farnebäck's algorithm [32]. In particular, we use the available implementation in OpenCV*[1]*, which is based on modeling the neighborhoods of each pixel of two consecutive frames by quadratic polynomials. This implementation allows a wide range of parameterizations, which are specified in Section 5* [19].

The resulting motion vectors, which are shown in Fig. B.10, are masked and quantized to produce weighted votes for local motion based on their magnitude, taking into account only those motion vectors that fall inside the G^{color} *grids. Such votes are locally accumulated into a* ν*-bin histogram over each grid cell according to the signed* ($0°$ - $360°$) *vector orientations. In contrast to HOG, HOF uses signed optical flow*

[1]This is an implementation of the work of [13], which can be found at `http://code.opencv.org`.

since the orientation information provides more discriminative power [19].

Depth modality

The grid cells in the depth modality G^{depth} are depth dense maps represented as planar images of pixels that measure depth values in millimeters. From this depth representation (projective coordinates) it is possible to obtain the "real world" coordinates by using the intrinsic parameters of the depth sensor. This new representation, which can be seen as a 3D point cloud structure $\mathcal{P}$, offers the possibility of measuring actual euclidean distances – those that can be measured in the real world [19].

After completing the former conversion, we propose to compute the surface normals for each particular point cloud $\mathcal{P}_{rij}$ (representing an arbitrary grid cell $\mathbf{G}^{\text{depth}}_{rij}$) and their distribution of angles summarized in a δ-bin histogram that describes the cell from the depth modality point of view [19].

Histogram of oriented depth normals (HON). *In order to describe an arbitrary point cloud $\mathcal{P}_{rij}$, the surface normal vector for each 3D point must be computed first. The normal 3D vector at a given point $\mathbf{p} = (p_x, p_y, p_z) \in \mathcal{P}$ can be seen as a problem of determining the normal of a 3D plane tangent to $\mathbf{p}$. A plane is represented by the origin point $\mathbf{q}$ and the normal vector $\mathbf{n}$. From the neighboring points $\mathcal{K}$ of $\mathbf{p} \in \mathcal{P}$, we first set $\mathbf{q}$ to be the average of those points* [19]:

$$\mathbf{q} \triangleq \bar{\mathbf{p}} = \frac{1}{|\mathcal{K}|} \sum_{\mathbf{p} \in \mathcal{K}} \mathbf{p}. \tag{B.8}$$

The solution of $\mathbf{n}$ can be then approximated as the smallest eigenvector of the covariance matrix $C \in \mathbb{R}^{3\times3}$ of the points in $\mathcal{P}^{\mathcal{K}}_{\mathbf{p}}$ [19].

The sign of $\mathbf{n}$ can be either positive or negative, and it cannot be disambiguated from the calculations. We adopt the convention of consistently re-orienting all computed normal vectors towards the depth sensor's viewpoint direction $\mathbf{z}$. Moreover, a neighborhood radius parameter determines the cardinality of $\mathcal{K}$, i.e. the number of points used to compute the normal vector in each of the points in $\mathcal{P}$. The computed normal vectors over a human body region is shown in Fig. B.10. Points are illustrated in white, whereas normal vectors are red lines (instead of arrows to ease the visualization). The next step is to build the histogram describing the distribution of the normal vectors' orientations [19].

A normal vector is expressed in spherical coordinates using three parameters: the radius, the inclination θ, and the azimuth φ. In our case, the radius is a constant value, so this parameter can be omitted. Regarding θ and φ, the cartesian-to-spherical coordinate transformation is calculated as: [19]

$$\theta = \arctan\left(\frac{n_z}{n_y}\right), \quad \varphi = \arccos\frac{\sqrt{(n_y^2 + n_z^2)}}{n_x}. \tag{B.9}$$

Therefore, a 3D normal vector can be characterized by a pair (θ, φ) and the depth description of a cell consists of a pair of δ_θ-bin and δ_φ-bin histograms (such that $\delta = \delta_\theta + \delta_\varphi$), L1-normalized and concatenated, describing the two angular distributions of the body surface normals within the cell [19].

Thermal modality

Whereas neither raw values of color intensity nor depth values of a pixel provide especially meaningful information for the human detection task, raw values of thermal intensity on their own are much more informative.

Histogram of thermal intensities and oriented gradients (HIOG). *The descriptor obtained from a cell in the thermal cue $\mathbf{G}_{rij}^{\text{thermal}}$ is the concatenation of two histograms. The first one is a histogram summarizing the thermal intensities, which spread across the interval $[0, 255]$. The second histogram summarizes the orientations of thermal gradients. Such gradients, computed by convolving a first derivative kernel in both directions, are binned in a histogram weighted by their magnitude. Finally, the two histograms are L1-normalized and concatenated. We used α_{i} bins for the intensities and α_{g} bins for the gradients' orientations* [19].

4.4 Uni-modal (description-level) classification

Since we wish to segment human body regions, we need to distinguish those from the other foreground regions segmented by the background subtraction algorithm. One way to tackle this task is from an uni-modal perspective [19].

From the previous step, each grid cell has been described using each and every description in $\mathcal{D}$. For the purpose of classification, we train a Gaussian Mixture Model for every cell (i, j) and description in $\mathcal{D}$. For a particular description d, we thereby obtain the set of GMM models $\mathcal{M}^{(d)} = \{\mathcal{M}_{ij}^{(d)} \mid \forall i \in \{1, ..., n\}, \forall j \in \{1, ..., m\}\}$ [19].

For predicting a new unseen region r to be either a subject or an object according to d, it is first partitioned into G_r, the cells' contents $\{\mathbf{G}_{rij}^{\omega(d)}\}_{\forall i,j}$ are described, and the $n \times m$ feature vectors representing the region in the d-space, $\{\mathbf{d}_{rij}^{(d)}\}_{\forall i,j}$, are evaluated in the corresponding mixtures' PDFs. The log-likelihood value associated with the (i, j)-th feature vector, $\mathbf{d}_{rij}^{(d)}$, is thus the one in the most probable component in the mixture $\mathcal{M}_{ij}^{(d)}$. Formally, we denote this log-likelihood value as $\ell_{rij}^{(d)}$. Eventually, the category – either subject or object – of the (i, j) cell according to d can be predicted by comparing the standardized log-likelihood $\hat{\ell}_{rij}^{(d)}$ with an experimentally selected threshold value $\tau_{ij}^{(d)}$.

However, given that we can have a different category prediction for each

cell, we first need to reach a consensus among cells. In order to do this, we convert the standardized log-likelihoods to confidence-like terms. This transformation consists of centering $\{\hat{\ell}_{rij}^{(d)} \mid \forall r \in R\}$ to $\tau_{ij}^{(d)}$ and scaling the centered values by a deviation-like term that is simply the mean squared difference in the sample with respect to $\tau_{ij}^{(d)}$. This way, we eventually come up with the confidence-like terms $\{\varrho_{rij}^{(d)} \mid \forall r \in R\}$ that conveniently differ in their sign depending on the category label: a negative sign for objects and a positive one for subjects; thus, the more negative (or positive) the value is, the more confidently we can categorize it as an object (or a subject).

Finally, the consensus among the cells of a certain region r can be attained by a voting scheme. For this purpose, we define the grid consensus function $g(r;d)$ as follows:

$$v_r^{(d,-)} = \sum_{i,j} \mathbb{1}\{\varrho_{rij}^{(d)} < 0\}\ ,\quad v_r^{(d,+)} = \sum_{i,j} \mathbb{1}\{\varrho_{rij}^{(d)} > 0\} \tag{B.10}$$

$$\bar{\varrho}_r^{(d,-)} = \frac{1}{v_r^{(d,-)}} \sum_{(i,j)\,|\,\varrho_{rij}^{(d)}<0} \varrho_{rij}^{(d)}\ , \tag{B.11}$$

$$\bar{\varrho}_r^{(d,+)} = \frac{1}{v_r^{(d,+)}} \sum_{(i,j)\,|\,\varrho_{rij}^{(d)}>0} \varrho_{rij}^{(d)} \tag{B.12}$$

$$g(r;d) = \begin{cases} 0 & \text{if } v_r^{(d,-)} > v_r^{(d,+)} \\ \mathbb{1}\left\{|\bar{\varrho}_r^{(d,-)}| < |\bar{\varrho}_r^{(d,+)}|\right\} & \text{if } v_r^{(d,-)} = v_r^{(d,+)} \\ 1 & \text{if } v_r^{(d,-)} < v_r^{(d,+)} \end{cases}\ , \tag{B.13}$$

where $v_r^{(d,-)}$ and $v_r^{(d,+)}$ keep count of the votes of the r grid cells for object (negative confidence) and subject (positive confidence), respectively. $\bar{\varrho}_r^{(d,-)}$ and $\bar{\varrho}_r^{(d,+)}$ are the averages of negative and positive confidences, respectively. In the case of a draw, the magnitude of the mean confidences obtained for both categories are compared. Since confidence values ϱ are centered at the decision threshold τ, these can be interpreted as a margin distance. From these calculations, the cells' decisions can be aggregated and the category of a grid r determined from each of the descriptions' point of view.

4.5 Multi-modal fusion

Our hypothesis is that the fusion of different modalities and descriptors, potentially providing a more informative and richer representation of the scenario, can improve the final segmentation result [19].

Learning-based fusion approach

As before, the category of a grid r should be predicted. However, instead of just relying on individual descriptions, we exploit the confidences ϱ provided by the GMMs in the different cells and types of description altogether. *This approach follows the Stacked Learning scheme [24, 72], which involves training a new learning algorithm by combining previous predictions obtained with other learning algorithms* [19]. More precisely, each grid r is represented by a vector $\mathbf{v}_r$ of confidences:

$$\mathbf{v}_r = (\varrho_{r11}^{(1)}, ..., \varrho_{rNM}^{(1)}, ..., \varrho_{r11}^{(|\mathcal{D}|)}, ..., \varrho_{rNM}^{(|\mathcal{D}|)}, y_r), \tag{B.14}$$

where y_r is the actual category of the r grid. Using such representation of the confidences in the different grid cells and modalities, we build a data sample containing the R feature vectors of this kind. In this way, any supervised learning algorithm can be used to learn from these data and infer more reliable predictions than using individual descriptions and defined voting scheme for cells' consensus. For this purpose, we use a Random Forest classifier [15] after an experimental evaluation of different state-of-the-art classifiers.

5 Evaluation

We test our approach in the novel RGB-D-T dataset and compare it to other state-of-the-art approaches. First we detail the experimental methodology and evaluation parameters and then provide the experiments' results and a discussion about them.

5.1 Experimental methodology and validation measures

We divided the dataset into 10 continuous sequences, as listed in Table B.3, and performed a leave-one-sequence-out cross-validation so as to compute the out-of-sample segmentation overlap. The unequal length of the sequences stems from the posture variability criterion followed: to ensure that very similar postures are not repeated in the different folds (*i.e.* sequences).

In addition, we performed a model selection in each training partition in order to find the optimal values for the GMMs' experimental parameters: k (number of components in the mixture), τ (decision threshold), and ϵ (stopping criterion for fitting the mixtures). We provide more detailed information about their values in Section 5.2. Although we used the leave-one-sequence-out cross-validation strategy again, we applied it this time to the remaining $N-1$ training sequences. In each inner fold, a grid search was carried out to measure the performance of each combination (k, τ, ϵ). The optimal combination, *i.e.*, the one that showed the best average across the 10×9 model selections, was

Sequence id.	Scene id.	No. frames	Start-end frame
1	1	134	00001-00134
2		905	00135-01638
3		762	01639-02400
4	2	247	00001-00247
5		816	00248-01063
6		463	01064-01526
7		690	01527-02216
8	3	142	00001-00142
9		848	00143-01449
10		951	01450-02400

Table B.3: Division of the scenes into 10 sequences (or partitions) of different length.

used to train the final model eventually validated in the corresponding test sequence.

The parameters of the supervised classifiers in the learning-based fusions were selected following the same validation procedure as above but considered the vectors of stacked confidences instead of the original descriptors. While the selection of k, τ, and ϵ was sufficiently exhaustive, given their nature, the parameters involved in these supervised learning algorithms often require more exhaustive searches to fine-tune their values. In order to find the best parameters while keeping the number of combinations manageable, we performed a two-level grid search, which consisted of a first coarse grid search followed by a second narrow grid search around the coarse optimal values.

As previously mentioned, we computed an overlap measure in order to evaluate the performance of our baseline. The overlap was first computed per person-ID and frame, and then averaged across all IDs in that frame. For the computation, we used intersection-over-union $\frac{|A \cap B|}{|A \cup B|}$, where A is a ground-truth region with a certain person-ID and B the region of prediction with its pixels coinciding with those of A. *Having computed the overlaps at frame-level, the overlap of a sequence is thereby calculated as the mean overlap of all those frames containing at least one blob, whether it be in the ground-truth or in the prediction mask* [19].

As stated in Section 4.1, the depth cue suffers from a halo effect around people or objects, thus complicating an accurate pixel-level segmentation at blob contours when applying background subtraction. This lack of accuracy is also caused by possible distortions, noise, or other problems, and decreases the final overlap. To tackle this problem, a do not care region *(DCR) is often used* [19]. A DCR simply defines a border region of pixels over the silhouette contours in both the prediction and contour masks that are not taken into account for the overlap computation. *In this way, we can compare the effect of using a growing DCR to the actual overlap* [19].

5.2 Parameters and settings

We experimentally set $\lambda_1 = 0.1$ and $\lambda_2 = 0.6$ for the automatic tagging of regions of interest. *We also set* $\sigma_{otsu} = 8.3$ *for a connected component area of at least 0.1% of the image and* $\sigma_{otsu} = 12$ *for other cases* [19]. These settings were established in order to maintain a trade-off between finding the maximum number of overlapping people situations without dividing a subject in different regions, depending on the variation of depth of the body parts.

Since it is not possible to have a pixel-to-pixel correspondence among modalities, we define the correspondence at a grid cell level. The grids have been partitioned in $m \times n$ *cells, with* $m = 2$ *and* $n = 2$ [19].

For the HOG descriptor, each grid cell was resized to 64×128 *pixels and divided in rectangular blocks of* 32×32 *pixels, which were, in turn, divided into rectangular local spatial regions of* 16×16 *pixels. We also set* $\kappa = 9$*. The information of each local spatial region is concatenated, resulting in a vector of 36 values per HOG-block. This brings the final vector size of a grid cell to 4 HOG-blocks vertically* $\times$ *2 HOG-blocks horizontally* $\times$ *4 HOG-cells per block* $\times$ *9 bins per HOG-cell, making a total of 288 components/dimensions* [19]. To further reduce the vector length and avoid the curse of dimensionality, we applied PCA to such vector, retaining 95% of the information. This way, the number of components of the feature vectors from all descriptions do not differ greatly.

In order to compute optical flow, we fixed the parameters of the given implementation based on the best-performing ones from the tests performed in [16]. *Specifically, we set the average window size to 2, the size of the pixel neighborhood considered when finding polynomial expansion in each pixel to 5, and the standard deviation of the Gaussian that is used to smooth derivatives used as a basis for the polynomial expansion to 1.1. The remaining parameters were set to their default values. For the motion descriptor (HOF), we defined* $\nu = 8$ *to produce an 8-dimensional feature vector* [19].

For the depth descriptors (HON), we defined $\delta_\theta = 8$ *and* $\delta_\varphi = 8$*, whereas for the thermal descriptors (HIOG), we defined* $v_{\mathrm{i}} = 8$ *and* $v_{\mathrm{g}} = 8$*, as they are standard values often used in the literature* [19].

In the GMM-related experiments, we set $k = \{2, 4, 6, 8, 10, 12\}$ and $\tau = \{-3, -2.5, -2, -1.5, -1.25, -1, -0.75, -0.5, -0.4, \ldots, 0.5, 0.75, 1, 1.25, 1.5, 2, 2.5, 3\}$. In order to avoid overfitting problems, we also optimized the termination criterion of the Expectation-Maximization algorithm used for training the GMMs, $\epsilon = \{1e-2, 1e-3, 1e-4, 1e-5\}$.

Among many existing state-of-the-art supervised learning algorithms able to perform the fusion, we tested the following: Adaptive Boosting, Multi-Layer Perceptron (with both sigmoidal and radial basis activation functions), Support Vector Machines (linear and radial basis function kernels), and Random Forest. In the AdaBoost experiment, we selected the number of possible weak classifiers and the weight trimming rates among $\{10, 20, 50, 100, 200, 500, 1000\}$

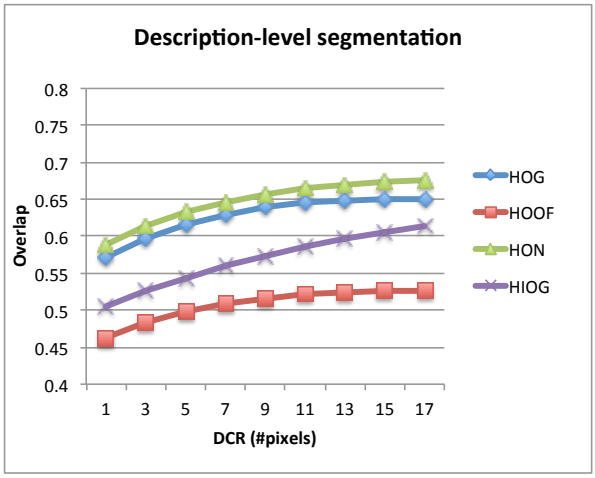


Fig. B.11: Results obtained from the different individual descriptions (HOG, HOF, HON, and HIOG) in terms of overlap.

and $\{0, 0.7, 0.75, 0.8, \ldots, 1\}$, respectively; in the MLP, we chose the number of neurons of the hidden layer among $\{2, 5, 10, 15, \ldots, 50, 60, 70, \ldots, 100\}$; in the SVM, we tested the regularization and the gamma parameters within $\{1e-7, 1e-6, \ldots, 1e4\}$ and in $\{1e-7, 1e-6, \ldots, 1e2\}$; and finally, in the RF we selected the maximum depth of the trees from $\{2, 4, 8, 16, 32, 64\}$, the maximum number of trees from $\{1, 2, 4, 8, 16, 32, 64, 128\}$, and the proportion of random variables to consider in node split from $\{0.05, 0.1, 0.2, 0.4, 0.8, 1\}$.

Regarding the DCR size, we tested several values (number of pixels) in the interval $[2 \cdot 0 + 1, \ldots, 2 \cdot 8 + 1]$.

In addition, and to better capture the posture variability, we augmented the training data by including the mirrored versions of the regions of interest along the vertical axis, as well as the original ones. Nonetheless, at the test stage, we considered only original regions of interest.

5.3 Experiments

In this section, we illustrate the performance of our baseline in terms of overlap after carrying out an extensive experiment. First, we illustrate the performance of the different descriptions (HON, HOF, HON, and HIOG). Second, we compare the best description to the learning-based fusions. Third, we show the performance of the baseline in the different sequences (test partitions). Fourth, we compare the evaluation of the baseline using the color/depth ground-truth masks vs. the thermal ones. And fifth, we compare our baseline to two standard techniques of the state of the art performing segmentation in the different modalities. In all cases we measure the overlap

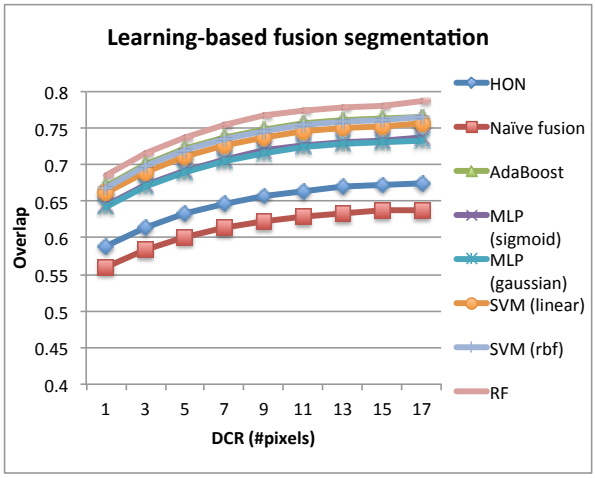


Fig. B.12: Results obtained from the best individual descriptions (HON), a naive fusion, and different learning-based fusions, in terms of overlap.

in function of the DCR size and compare it to color/depth ground-truth masks, unless otherwise stated.

Experiment: HOG, HOF, HON, and HIOG descriptions

We evaluated the performance of the proposed descriptions (HOG, HOF, HON, and HIOG) when predicting on their own. The overlap results shown in Fig. B.11, where each descriptor overlap index is computed with respect to their specific modality ground-truth masks, demonstrate the superior performance of the HON descriptor computed in the depth modality, which reach 67.5% of overlap and improve by 14% (on average for the different DCR sizes) the results of the worst performing description. The HOG description in the color modality came in a close second (65%), achieving 2.5% less overlap than HON (in average). The worst results were obtained by the motion cue in this case, probably because they were uninformative when dealing with static postures, which are abundant in our data. Despite this, it is able to segment people while achieving more than 50% of such a pessimistic measure as overlap. Note, also, the different upward trend of HIOG in the thermal modality. We discuss this phenomenon, which is due to the color-to-thermal registration, in Section 5.4.

Experiment: learning-based fusion

In the second experiment, we compared the learning-based fusion with different classifiers against both the best performing description (HON) and a naive

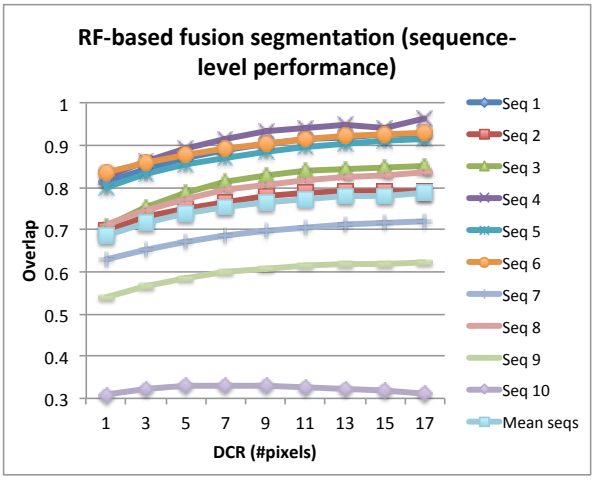


Fig. B.13: Results obtained from the RF-based fusion (the best learning-based fusion) in terms of overlap for the different sequences.

fusion we designed in order to give more credit to the better performance of the learning-based fusions. The naive fusion simply averages the cells confidences along the different modalities and then aggregates the averaged cell confidences as described in Section 4.4.

Fig. B.12 shows that the better performing method was the Random Forest classifier (up to 78.6% of overlap), which thus became our choice for the baseline. This supposed an improvement over HON of 10% (on average). On the other hand, the worst performing fusion (MLP with gaussian activation function) also presented an improvement over HON, but only of 5% (on average).

The naive fusion resulted in an overlap of 63.9%, which was substantially lower than both HON and HOG.

Once the best classifier for the learning-based fusion was determined, we measured separately the performance of our baseline on the different sequences. Fig. B.13 depicts the performance in the sequences. Notice that there is a large difference in performance across the evaluated sequences. Four of them – *Seq.1*, *Seq.4*, *Seq.5*, and *Seq.6* – exhibit saturation on the improvement of performance around 90% at DCR of 11-13 pixels. Four others – *Seq.2*, *Seq.3*, *Seq.7*, and *Seq.8* – are closer to the mean performance *Mean seqs*. And two of them – *Seq.9* and *Seq.10* – suffer a more severe drop in performance, especially *Seq.10*. We discuss plausible reasons for this further on in the paper.

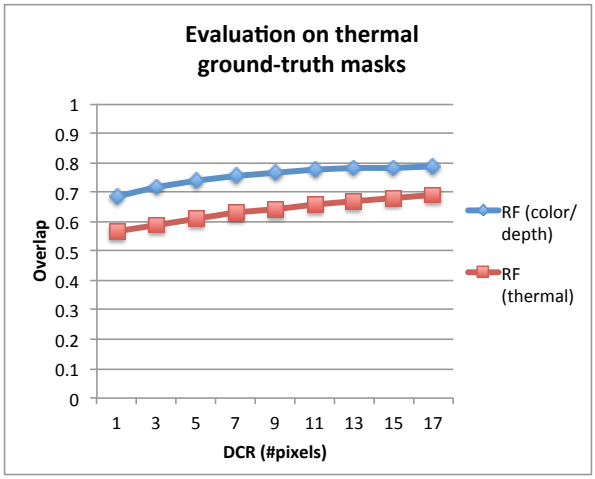


Fig. B.14: Comparison of performance measuring the overlap in the thermal registered masks against the manually annotated masks from color/depth.

Experiment: evaluation on thermal ground-truth masks

In addition, we measured the performance of our most successful approach on the thermal masks in order to quantitatively measure the decrease in performance caused by the misalignment in the thermal-to-color registration. Fig. B.14 reveals a relatively small decrease in performance. This fact somehow justifies the slightly poorer performance of HIOG in respect to HON and HOG, as previously depicted in Section 5.3, and why any thermal-related descriptors would pay a price when evaluated in the thermal ground-truth.

Experiment: comparison to state-of-the-art approaches

Since there is no approach that uses the three modalities for human body segmentation, we compared our baseline with two successful state-of-the-art approaches for such task performing in either the color or the depth cue.

One was the work of [17], which performs solely on the depth modality. This work, based on that of [81], describes depth pixels by a set of depth-invariant features generated from the normalized depth differences at pairs of random offsets in respect to the evaluated pixel. From this description, a Random Forest classifier is able to classify each pixel as a body part. In our experiments, we used the open-source implementation made available as part of the Point Cloud Library[2] along with a set of pre-trained trees[3]. In this way we were able to ensure that the method was not relying on tracking techniques

[2] `http://pointclouds.org/documentation/tutorials/gpu_people.php`
[3] `https://github.com/PointCloudLibrary/data/tree/master/people/results`

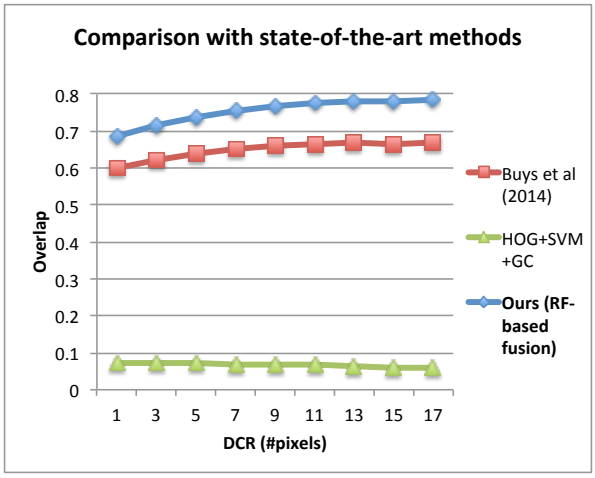


Fig. B.15: Comparison of our baseline (using RF-based fusion) with other state-of-the-art approaches that perform human body segmentation from color imagery (HOG+SVM+GC) and depth maps [17].

– for a fairer comparison to our approach – as would have been the case with the implementation of [81] found in the Kinect SDK[4]. Furthermore, we took advantage of the extracted foreground masks from Section 4.1 in order to apply the body part detector only to foreground pixels; this way, we avoided the apparition of false body part detections all around the scene.

We also compared our approach with that of HOG + SVM + GC (GrabCut) for people segmentation in the color modality. We used the OpenCV available implementations, which are based on the original algorithms [26, 75]. The HOG + SVM combination, in particular, detects people as bounding boxes, and the inner dense silhouettes are then segmented by means of GC. The latter is applied in an automatic fashion, learning the GMMs of 70% of the bounding box as *Probably Foreground* and the rest as *Probably Background*.

Both approaches were trained in independent but larger datasets that ensured more variation than if they had been trained in our dataset. As shown in Fig. B.15, our approach outperformed the other baselines when applied to our dataset.

Our baseline largely improved the HOG + SVM + GC approach. However, [17] achieved a result comparable to ours, with a maximum overlap of 67.1%. Despite that, our approach also improved this one by more than 10%.

[4] [81] specified in the "Acknowledgements" section that the tracking system of Kinect SDK was built based on the research they presented in the paper.

5.4 Discussion

The results we obtained showed that fusing different descriptions enhances the representation of the scene, thus increasing the final overlap when segmenting subjects and discriminating from other artifacts present in the scene [19].

Among the modalities included in our approach, we considered the thermal modality to be of great importance. One cannot guarantee human presence just because of large thermal intensity readings, since many non-human entities such as animals or unanimated objects can emit a considerable amount of heat. However, relatively low thermal intensities are, indeed, highly likely to imply the absence of human presence. This leads, in our case, to the classification of that region as a background category. Hence, in the context of human-background classification, we can consider this "human heat" prior a valuable piece of information that, used together with the thermal gradients and later fused with other cues, enhances the overall performance of our method. In Fig. B.16, we illustrated some situations in which the thermal contribution was of great importance to a proper segmentation. Nonetheless, we found the use of the modalities altogether to be very important for the segmentation task.

The set of simple yet reliable descriptions extracted from the multiple cues produced errors somehow uncorrelated. This could be seen in the qualitative results[5]. Our initial assumption was that the learning-based fusion should be able to take advantage of this lack of correlation and thus improve individual results. The quantitative results illustrated in Section 5.3 confirmed the validity of our initial assumption. The RF-based fusion, in particular, improved the individual descriptions by 25% on average when compared to HOF (the worst description) and 10% when comparing to HON (the best description). Moreover, the importance of the learning process in the fusion step was also assessed comparing the results of the learning-based approach to a more naive fusion of confidences.

The selection of the best classifier also proved to be crucial, doubling the improvement of performance with respect to HON when choosing RF over a MLP with gaussian activation function (from 5% to 10%). In fact, a SVM classifier with linear kernel performed surprisingly well, demonstrating the stacked vectors of confidences to be linearly separable features. Yet the RF classifier increased the overlap results 2.5% (on average) with respect to the linear SVM, showing that there was still room for improvement.

We also studied the performance of each of the sequences. In 7 out of 10 sequences, results were above the mean. The poor performance in one of them, *Seq. 10*, reduced the *Mean seqs* overlaps by almost 5% (on average). After checking the predicted masks, we noticed a false positive on a chair's back

[5]Check the video included as supplementary material in which some qualitative results are shown, named `trimodal_seg_results.mp4`.

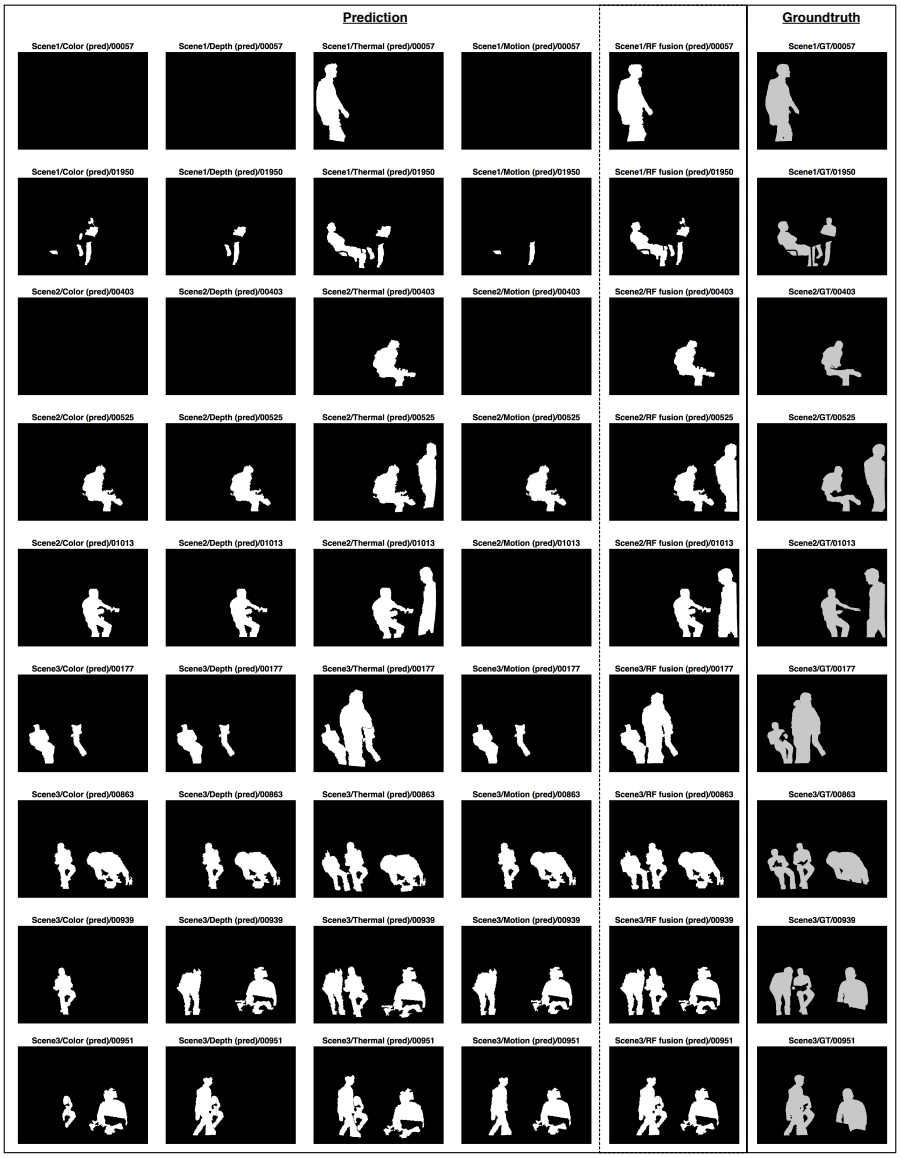


Fig. B.16: Qualitative results illustrating the importance of the thermal cue, with each row representing a frame. For each frame, we show the human prediction masks obtained from the different descriptions separately, in addition to the prediction from the fusion approach using a Random Forest classifier. From left to right, the predictions using: Color (HOG), Depth (HON), Thermal (HIOG), Motion (HOF), and RF-based fusion. The last column corresponds to the segmentation ground-truth mask. On top of each binary image, we indicate "sequence name"/"modality name" (or GT if ground-truth)/"frame ID".

region, which appeared quite static during the whole sequence and was a relatively big image region – because it was close to the camera. The difficulty level of this sequence can be better seen qualitatively in the last two rows of Fig. B.1. As mentioned before, this scenario contains wide windows with a large amount of sunlight, which may disturb the depth data. Moreover, the color of the subject's jumper is extremely similar to the color of the couch, making it difficult for the color modality. Another interesting effect is the heat mark that the subject bodies left on the couch in the thermal modality, which may be mistaken for a real subject.

Accurate pixel-level segmentation is a complex task in state-of-the-art techniques [19]. In these scenarios, a DCR is often considered. In our case, experiments showed marginal improvements for DCR sizes greater than 11 pixels, except for the case of thermal modality, which exhibited a particular upward trend. *It is important to note that thermal descriptions cannot reach overlap values as good as the other descriptions. The reason for this is that the binary masks* F^{thermal} *were created from* F^{depth} *using the registration algorithm, which cannot be accurate up to pixel level, in such a way that the ground-truth and registered masks differ slightly, especially on the left and right sides of the image* [19]. As one can observe, this misalignment caused by the registration algorithm introduced an additional error to the depth's halo effect, which kept being palliated with the biggest DCR sizes.

It is also worth discussing the causes of some misclassifications that we noticed. One of the problems originates at the beginning of the chain. Since background subtraction reduces the search space, it may reject some actual person parts. This happens mainly when a person is situated at the same depth as something that belongs to the background model. This could be improved by combining the different modalities in order to learn the background model. Furthermore, the contours of the foreground binary masks may not be perfect, either. One possible solution would be to apply GrabCut or other post-segmentation approaches to refine and smooth the contours, which in turn would improve segmentation accuracy. Another issue is that some regions considered unknown *– mostly those generated when one person overlaps other – differ considerably from those that are used to train the different models. Hence, the classification of such regions is not a trivial task* [19].

6 Conclusions

We first introduced a novel RGB-Depth-Thermal dataset of video sequences, which contains several subjects interacting with everyday objects [19], along with a registration algorithm and the manual pixel-level annotations of human masks. Second, we proposed a multi-modal human body segmentation approach using the registered RGB-Depth-Thermal data as a preprocessing step for human activity recognition tasks.

The registration algorithm registered the different data modalities using multiple homographies generated from several views of the proposed calibration device. The segmentation baseline segmented the people appearing in a set of 10 trimmed video sequences out of the three recorded scenes. *It consisted of, first, a non-adaptive background subtraction approach in order to extract the regions of interest* [19] that deviate from the depth-background model previously learned. The regions from the different modalities were partitioned in a grid of cells. The cell were then described in the corresponding modalities using state-of-the-art image feature descriptors. *HOG and HOF were computed on RGB color imagery, a histogram of intensity gradients on thermal, and histograms of normal vectors' orientations on depth* [19]. For each cell and modality, we modeled the distribution of descriptions using a GMM. During the prediction phase, cells were evaluated in the corresponding GMMs and the obtained likelihoods turned into confidence-like terms and stacked in a feature vector representation. A supervised learning algorithm, such as Random Forest, learned to categorize such representation into human or non-human regions.

In the end, we found notable performance improvements with the proposed learning-based fusion strategies in comparison to each isolated modality, and Random Forest obtained the best results. Furthermore, our baseline outperformed different state-of-the-art uni-modal segmentation methods, hence demonstrating the power of multi-modal fusion.

Acknowledgements

This work was partly supported by the Spanish project TIN2013-43478-P. The work of Albert Clapés was supported by SUR-DEC of the Generalitat de Catalunya and FSE. We would like to thank Anders Jørgensen for his valuable help in capturing the dataset.

References

[1] B. Abidi, "IRIS thermal/visible face database," DOE University Research Program in Robotics under grant DOE-DE-FG02-86NE37968, 2007.

[2] K. Alahari, G. Seguin, J. Sivic, I. Laptev, and Others, "Pose estimation and segmentation of people in 3D movies," in *ICCV 2013-IEEE Int. Conf. Comput. Vis.*, 2013.

[3] S. Alpert, M. Galun, R. Basri, and A. Brandt, "Image segmentation by probabilistic bottom-up aggregation and cue integration," in *Comput. Vis. Pattern Recognition, 2007. CVPR '07. IEEE Conf.*, 2007, pp. 1–8.

[4] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: people detection and articulated pose estimation," in *Comput. Vis. Pattern Recognition, 2009. CVPR 2009. IEEE Conf.*, 2009, pp. 1014–1021.

[5] ——, "Monocular 3D pose estimation and tracking by detection," in *Comput. Vis. Pattern Recognit. (CVPR), 2010 IEEE Conf.*, 2010, pp. 623–630.

[6] I. B. Barbosa, M. Cristani, A. Del Bue, L. Bazzani, and V. Murino, "Re-identification with RGB-D sensors," in *Comput. Vision–ECCV 2012. Work. Demonstr.* Springer, 2012, pp. 433–442.

[7] M. Bertozzi, A. Broggi, C. H. Gomez, R. I. Fedriga, G. Vezzoni, and M. Del Rose, "Pedestrian detection in far infrared images based on the use of probabilistic templates," in *Intell. Veh. Symp. 2007 IEEE*, 2007, pp. 327–332.

[8] J.-Y. Bouguet, "Camera calibration toolbox for matlab," 2004.

[9] L. Bourdev and J. Malik, "Poselets: body part detectors trained using 3D human pose annotations," in *Comput. Vision, 2009 IEEE 12th Int. Conf.*, 2009, pp. 1365–1372.

[10] T. Bouwmans, "Recent advanced statistical background modeling for foreground detection: a systematic survey," *RPCS*, vol. 4, no. 3, pp. 147–176, 2011.

[11] T. Bouwmans, F. El Baf, B. Vachon, and Others, "Background modeling using mixture of gaussians for foreground detection-a survey," *Recent Patents Comput. Sci.*, vol. 1, no. 3, pp. 219–237, 2008.

[12] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images," in *Comput. Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE Int. Conf.*, vol. 1, 2001, pp. 105–112.

[13] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'reilly, 2008.

[14] M. Bray, P. Kohli, and P. H. S. Torr, "Posecut: Simultaneous segmentation and 3D pose estimation of humans using dynamic graph-cuts," in *Comput. Vision–ECCV 2006*. Springer, 2006, pp. 642–655.

[15] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[16] K. Brkić, S. Rašić, A. Pinz, S. Šegvić, and Z. Kalafatić, "Combining spatio-temporal appearance descriptors and optical flow for human action recognition in video data," *arXiv Prepr. arXiv1310.0308*, 2013.

[17] K. Buys, C. Cagniart, A. Baksheev, T. De Laet, J. De Schutter, and C. Pantofaru, "An adaptable system for rgb-d based human body detection and pose estimation," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 39–52, 2014.

[18] M. Camplani and L. Salgado, "Background foreground segmentation with RGB-D Kinect data: an efficient combination of classifiers," *J. Vis. Commun. Image Represent.*, vol. 25, no. 1, pp. 122–136, 2014.

[19] C. P. Cantariño, "Tri-modal human body segmentation," *Unpublished Master Thesis, Universitat de Barcelona, Spain*, 2014.

[20] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: image segmentation using expectation-maximization and its application to image querying," *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 24, no. 8, pp. 1026–1038, 2002.

[21] J. Charles and M. Everingham, "Learning shape models for monocular human pose estimation from the Microsoft Xbox Kinect," in *Comput. Vis. Work. (ICCV Work. 2011 IEEE Int. Conf.*, 2011, pp. 1202–1208.

[22] S. Y. Chun and C.-S. Lee, "Applications of human motion tracking: smart lighting control," in *Comput. Vis. Pattern Recognit. Work. (CVPRW), 2013 IEEE Conf.*, 2013, pp. 387–392.

[23] A. Clapés, M. Reyes, and S. Escalera, "User identification and object recognition in clutter scenes based on RGB-Depth analysis," in *Articul. Motion Deform. Objects*. Springer, 2012, pp. 1–11.

[24] W. W. Cohen, "Stacked sequential learning," DTIC Document, Tech. Rep., 2005.

[25] C. Dai, Y. Zheng, and X. Li, "Pedestrian detection and tracking in infrared imagery using shape and appearance," *Comput. Vis. Image Underst.*, vol. 106, no. 2, pp. 288–299, 2007.

[26] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Comput. Vis. Pattern Recognition, 2005. CVPR 2005. IEEE Comput. Soc. Conf.*, vol. 1, 2005, pp. 886–893.

[27] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Comput. Vision–ECCV 2006*. Springer, 2006, pp. 428–441.

[28] J. W. Davis and V. Sharma, "Robust background-subtraction for person detection in thermal imagery," *IEEE Int. Wkshp. Object Track. Classif. Beyond Visible Spectr.*, 2004.

[29] ——, "Background-subtraction using contour-based fusion of thermal and visible imagery," *Comput. Vis. Image Underst.*, vol. 106, no. 2, pp. 162–182, 2007.

[30] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge 2012 results." See http://www.pascal-network. org/challenges/VOC/voc2012/workshop/index. html, 2012.

[31] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool, "Random forests for real time 3D face analysis," *Int. J. Comput. Vis.*, vol. 101, no. 3, pp. 437–458, 2013.

[32] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image Anal.* Springer, 2003, pp. 363–370.

[33] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 32, no. 9, pp. 1627–1645, 2010.

[34] A. Fernández-Caballero, J. C. Castillo, J. Serrano-Cuerda, and S. Maldonado-Bascón, "Real-time human segmentation in infrared videos," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2577–2584, 2011.

[35] E. J. Fernández-Sánchez, J. Díaz, and E. Ros, "Background subtraction based on color and depth using active sensors," *Sensors*, vol. 13, no. 7, pp. 8895–8915, 2013.

[36] S. Fidler, R. Mottaghi, A. Yuille, and R. Urtasun, "Bottom-up segmentation for top-down detection," in *Comput. Vis. Pattern Recognit. (CVPR), 2013 IEEE Conf.*, 2013, pp. 3294–3301.

[37] R. Gade, A. Jorgensen, and T. B. Moeslund, "Long-term occupancy analysis using graph-based optimisation in thermal imagery," in *Comput. Vis. Pattern Recognit. (CVPR), 2013 IEEE Conf.*, 2013, pp. 3698–3705.

[38] R. Gade and T. B. Moeslund, "Thermal cameras and applications: a survey," *Mach. Vis. Appl.*, vol. 25, no. 1, pp. 245–262, 2014.

[39] D. Giordano, S. Palazzo, and C. Spampinato, "Kernel density estimation using joint spatial-color-depth data for background modeling," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 4388–4393.

[40] R. B. Girshick, P. F. Felzenszwalb, and D. A. Mcallester, "Object detection with grammar models," in *Adv. Neural Inf. Process. Syst.*, 2011, pp. 442–450.

[41] G. Gordon, T. Darrell, M. Harville, and J. Woodfill, "Background estimation and removal based on range and color," in *Comput. Vis. Pattern Recognition, 1999. IEEE Comput. Soc. Conf. on.*, vol. 2, 1999.

[42] V. Gulshan, V. Lempitsky, and A. Zisserman, "Humanising grabCut: learning to segment humans using the Kinect," in *Comput. Vis. Work. (ICCV Work. 2011 IEEE Int. Conf.*, 2011, pp. 1127–1133.

[43] A. Hernández-Vela, M. A. Bautista, X. Perez-Sala, V. Ponce, X. Baró, O. Pujol, C. Angulo, and S. Escalera, "BoVDW: Bag-of-Visual-and-Depth-Words for gesture recognition," in *Pattern Recognit. (vICPR), 2012 21st Int. Conf.* IEEE, 2012, pp. 449–452.

[44] A. Hernández-Vela, N. Zlateva, A. Marinov, M. Reyes, P. Radeva, D. Dimov, and S. Escalera, "Graph cuts optimization for multi-limb human segmentation in depth maps," in *Comput. Vis. Pattern Recognit. (CVPR), 2012 IEEE Conf.*, 2012, pp. 726–732.

[45] R. I. Hg, P. Jasek, C. Rofidal, K. Nasrollahi, T. B. Moeslund, and G. Tranchet, "An RGB-D database using Microsoft's Kinect for Windows for face detection," in *Signal Image Technol. Internet Based Syst. (SITIS), 2012 Eighth Int. Conf.* IEEE, 2012, pp. 42–46.

[46] B. Holt, E.-J. Ong, H. Cooper, and R. Bowden, "Putting the pieces together: Connected poselets for human pose estimation," in *Comput. Vis. Work. (ICCV Work. 2011 IEEE Int. Conf.*, 2011, pp. 1196–1201.

[47] T. Huynh, R. Min, and J.-L. Dugelay, "An efficient LBP-based descriptor for facial depth images applied to gender recognition using RGB-D face data," in *Comput. Vision-ACCV 2012 Work.* Springer, 2013, pp. 133–145.

[48] R. Irani, K. Nasrollahi, M. Oliu, C. Corneanu, S. Escalera, C. Bahnsen, D. Lundtoft, T. B. Moeslund, T. Pedersen, M.-L. Klitgaa, and L. Petrini, "Spatiotemporal analysis of rgb-d-t facial images for multi-modal pain level recognition," *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2015.

[49] H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from RGB-D videos," *Int. J. Rob. Res.*, vol. 32, no. 8, pp. 951–970, 2013.

[50] M. P. Kumar, P. H. S. Ton, and A. Zisserman, "Obj cut," in *Comput. Vis. Pattern Recognition, 2005. CVPR 2005. IEEE Comput. Soc. Conf.*, vol. 1, 2005, pp. 18–25.

[51] L. Ladický, P. Sturgess, K. Alahari, C. Russell, and P. H. S. Torr, "What, where and how many? combining object detectors and crfs," in *Comput. Vision–ECCV 2010*. Springer, 2010, pp. 424–437.

[52] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *Work. Stat. Learn. Comput. Vision, ECCV*, vol. 2, no. 5, 2004, p. 7.

[53] ——, "Robust object detection with interleaved categorization and segmentation," *Int. J. Comput. Vis.*, vol. 77, no. 1-3, pp. 259–289, 2008.

[54] A. Levin and Y. Weiss, "Learning to combine bottom-up and top-down segmentation," in *Comput. Vision–ECCV 2006*. Springer, 2006, pp. 581–594.

[55] A. Leykin and R. Hammoud, "Robust multi-pedestrian tracking in thermal-visible surveillance videos," in *Comput. Vis. Pattern Recognit. Work. 2006. CVPRW'06. Conf.* IEEE, 2006, p. 136.

[56] A. Leykin, Y. Ran, and R. Hammoud, "Thermal-visible video fusion for moving target tracking and pedestrian classification," in *Comput. Vis. Pattern Recognition, 2007. CVPR'07. IEEE Conf.*, 2007, pp. 1–8.

[57] Z. Lin, L. S. Davis, D. Doermann, and D. DeMenthon, "An interactive approach to pose-assisted and appearance-based segmentation of humans," in *Comput. Vision, 2007. ICCV 2007. IEEE 11th Int. Conf.*, 2007, pp. 1–8.

[58] O. Lopes, M. Reyes, S. Escalera, and J. Gonzalez, "Spherical blurred shape model for 3d object and pose recognition: Quantitative analysis and hci applications in smart environments," 2014.

[59] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Comput. Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE Int. Conf.*, vol. 2, 2001, pp. 416–423.

[60] A. Mittal, L. Zhao, and L. S. Davis, "Human body pose estimation using silhouette shape analysis," in *Proceedings. IEEE Conf. Adv. Video Signal Based Surveillance, 2003.*, 2003, pp. 263–270.

[61] T. B. Moeslund, *Visual analysis of humans: looking at people*. Springer, 2011.

[62] A. Møgelmose, C. Bahnsen, T. Moeslund, A. Clapés, and S. Escalera, "Tri-modal person re-identification with rgb, depth and thermal features," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, June 2013, pp. 301–307.

[63] G. Mori, X. Ren, A. A. Efros, and J. Malik, "Recovering human body configurations: combining segmentation and recognition," in *Comput. Vis. Pattern Recognition, 2004. CVPR 2004. Proc. 2004 IEEE Comput. Soc. Conf.*, vol. 2, 2004, pp. II—-326.

[64] A. T. Nghiem, F. Bremond, M. Thonnat, and V. Valentin, "ETISEO, performance evaluation for video surveillance systems," in *Adv. Video Signal Based Surveillance, 2007. AVSS 2007. IEEE Conf.*, 2007, pp. 476–481.

[65] O. Nikisins, K. Nasrollahi, M. Greitans, and T. Moeslund, "Rgb-d-t based face recognition," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, Aug 2014, pp. 1716–1721.

[66] D. Olmeda, A. de la Escalera, and J. M. Armingol, "Contrast invariant features for human detection in far infrared images," in *Intell. Veh. Symp. (IV), 2012 IEEE*, 2012, pp. 117–122.

[67] O. Oreifej and Z. Liu, "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences," in *Comput. Vis. Pattern Recognit. (CVPR), 2013 IEEE Conf.*, 2013, pp. 716–723.

[68] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[69] H. Pirsiavash and D. Ramanan, "Steerable part models," in *Comput. Vis. Pattern Recognit. (CVPR), 2012 IEEE Conf.*, 2012, pp. 3226–3233.

[70] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun, "Real-time identification and localization of body parts from depth images," in *Robot. Autom. (ICRA), 2010 IEEE Int. Conf.*, 2010, pp. 3108–3113.

[71] R. Poppe, "A survey on vision-based human action recognition," *Image Vis. Comput.*, vol. 28, pp. 976–990, 2010.

[72] E. Puertas, S. Escalera, and O. Pujol, "Generalized multi-scale stacked sequential learning for multi-class classification," *Pattern Anal. Appl.*, pp. 1–15, 2013.

[73] N. Pugeault and R. Bowden, "Spelling it out: Real-time asl fingerspelling recognition," in *Comput. Vis. Work. (ICCV Work. 2011 IEEE Int. Conf.*, 2011, pp. 1114–1119.

[74] D. Ramanan, "Learning to parse images of articulated bodies," in *Adv. Neural Inf. Process. Syst.*, 2006, pp. 1129–1136.

[75] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: interactive foreground extraction using iterated graph cuts," in *ACM Trans. Graph.*, vol. 23, no. 3. ACM, 2004, pp. 309–314.

[76] T. Scharwächter, M. Enzweiler, U. Franke, and S. Roth, "Efficient multi-cue scene segmentation," in *Pattern Recognit.* Springer, 2013, pp. 435–445.

[77] L. A. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab, "Estimating human 3D pose from time-of-flight images based on geodesic distances and optical flow," in *Autom. Face & Gesture Recognit. Work. (FG 2011), 2011 IEEE Int. Conf.*, 2011, pp. 700–706.

[78] G. Sheasby, J. Valentin, N. Crook, and P. Torr, "A robust stereo prior for human segmentation," in *Comput. Vision–ACCV 2012*. Springer, 2013, pp. 94–107.

[79] G. Sheasby, J. Warrell, Y. Zhang, N. Crook, and P. H. S. Torr, "Simultaneous human segmentation, depth and pose estimation via dual decomposition," in *Br. Mach. Vis. Conf. Student Work. BMVW*, 2012.

[80] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 22, no. 8, pp. 888–905, 2000.

[81] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1297–1304. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2011.5995316

[82] L. Spinello and K. O. Arras, "People detection in RGB-D data," in *Intell. Robot. Syst. (IROS), 2011 IEEE/RSJ Int. Conf.*, 2011, pp. 3838–3843.

[83] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Comput. Vis. Pattern Recognition, 1999. IEEE Comput. Soc. Conf. on.*, vol. 2, 1999.

[84] M. Stefańczyk and W. Kasprzak, "Multimodal segmentation of dense depth maps and associated color information," in *Comput. Vis. Graph.* Springer, 2012, pp. 626–632.

[85] F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi, "Pedestrian detection using infrared images and histograms of oriented gradients," in *Intell. Veh. Symp. 2006 IEEE*, 2006, pp. 206–212.

[86] L. Susperregi, J. M. Martínez-Otzeta, A. Ansuategui, A. Ibarguren, and B. Sierra, "RGB-D, laser and thermal sensor fusion for people following in a mobile robot." *Int. J. Adv. Robot. Syst.*, vol. 10, 2013.

[87] A. Teichman and S. Thrun, "Learning to segment and track in RGB-D," in *Algorithmic Found. Robot. X*. Springer, 2013, pp. 575–590.

[88] S. Vidas, R. Lakemond, S. Denman, C. Fookes, S. Sridharan, and T. Wark, "A mask-based approach for the geometric calibration of thermal-infrared cameras," *Instrumentation and Measurement, IEEE Transactions on*, vol. 61, no. 6, pp. 1625–1635, June 2012.

[89] V. Vineet, G. Sheasby, J. Warrell, and P. H. S. Torr, "PoseField: An efficient mean-field based method for joint estimation of human pose, segmentation, and depth," in *Energy Minimization Methods Comput. Vis. Pattern Recognit.* Springer, 2013, pp. 180–194.

[90] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *Int. J. Comput. Vis.*, vol. 63, no. 2, pp. 153–161, 2005.

[91] L. Wang, Y. Qiao, and X. Tang, "Motionlets: mid-level 3D parts for human motion recognition," in *Comput. Vis. Pattern Recognit. (CVPR), 2013 IEEE Conf.*, 2013, pp. 2674–2681.

[92] W. Wang, J. Zhang, and C. Shen, "Improved human detection and classification in thermal images," in *Image Process. (ICIP), 2010 17th IEEE Int. Conf.*, 2010, pp. 2313–2316.

[93] Y. Wang, D. Tran, and Z. Liao, "Learning hierarchical poselets for human parsing," in *Comput. Vis. Pattern Recognit. (CVPR), 2011 IEEE Conf.*, 2011, pp. 1705–1712.

[94] T. Windheuser, U. Schlickewei, F. R. Schmidt, and D. Cremers, "Geometrically consistent elastic matching of 3D shapes: a linear programming solution," in *Comput. Vis. (ICCV), 2011 IEEE Int. Conf.*, 2011, pp. 2134–2141.

[95] C. Wolf, J. Mille, E. Lombardi, O. Celiktutan, M. Jiu, M. Baccouche, E. Dellandréa, C.-E. Bichot, C. Garcia, and B. Sankur, "The LIRIS human activities dataset and the ICPR 2012 human activities recognition and localization competition," in *LIRIS Umr 5205 CNRS/INSA Lyon/Universite'Claude Bernard Lyon 1/Universite'Lumie 're Lyon 2/E'cole Cent.*, 2012.

[96] L. Xia, C.-C. Chen, and J. K. Aggarwal, "Human detection using depth information by kinect," in *Comput. Vis. Pattern Recognit. Work. (CVPRW), 2011 IEEE Comput. Soc. Conf.*, 2011, pp. 15–22.

[97] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," in *Comput. Vis. Pattern Recognit. (CVPR), 2011 IEEE Conf.*, 2011, pp. 1385–1392.

[98] ——, "Articulated human detection with flexible mixtures of parts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 12, pp. 2878–2890, Dec 2013.

[99] B. Yao and L. Fei-Fei, "Grouplet: a structured image representation for recognizing human and object interactions," in *Comput. Vis. Pattern Recognit. (CVPR), 2010 IEEE Conf.*, 2010, pp. 9–16.

[100] L. Zhang, B. Wu, and R. Nevatia, "Pedestrian detection in infrared images based on local shape features," in *Comput. Vis. Pattern Recognition, 2007. CVPR'07. IEEE Conf.*, 2007, pp. 1–8.

[101] J. Zhao and S. C. Sen-ching, "Human segmentation by geometrically fusing visible-light and thermal imageries," *Multimed. Tools Appl.*, pp. 1–29, 2012.

[102] L. Zhu, Y. Chen, Y. Lu, C. Lin, and A. Yuille, "Max margin and/or graph learning for parsing the human body," in *Comput. Vis. Pattern Recognition, 2008. CVPR 2008. IEEE Conf.*, 2008, pp. 1–8.

[103] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Pattern Recognition, 2004. ICPR 2004. Proc. 17th Int. Conf.*, vol. 2. IEEE, 2004, pp. 28–31.

Paper C

Comparison of Multi-shot Models for Short-term Re-identification of People using RGB-D Sensor

Andreas Møgelmose, Chris Bahnsen, Thomas B. Moeslund

The paper has been published in the
Proceedings of the 10th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP), pp. 244-254, 2015.

The layout has been revised.

Abstract

This work explores different types of multi-shot descriptors for re-identification in an on-the-fly enrolled environment using RGB-D sensors. We present a full re-identification pipeline complete with detection, segmentation, feature extraction, and re-identification, which expands on previous work by using multi-shot descriptors modeling people over a full camera pass instead of single frames with no temporal linking. We compare two different multi-shot models; mean histogram and histogram series, and test them each in 3 different color spaces. Both histogram descriptors are assisted by a depth-based pruning step where unlikely candidates are filtered away. Tests are run on 3 sequences captured in different circumstances and lighting situations to ensure proper generalization and lighting/environment invariance.

1 Introduction

The task of person re-identification is about recognizing people that have been captured earlier by a camera in a surveillance network. The network may consist of one or more cameras, and can be placed in traditional surveillance contexts or more narrowly scoped areas, such as keeping track of a single queue of people. The objective is simple: When a person enters the field of view of a camera in the system, it must be determined whether or not this person has been seen before. Person re-identification is closely related to person tracking and person recognition. However, is has several extra challenges, that makes it less straight-forward [8]:

- There is no fully known gallery dataset. As opposed to traditional person recognition, the system must enroll new people on-the-fly, without them taking any action.
- Methods must be robust to pose changes. Since subjects are not required to participate actively, there are only weak constraints on pose and viewing angles.
- Sensor resolution is a big challenge. People simply passing by at various distances are to be re-identified, so it is not reasonable to use hard biometrics like fingerprints or face recognition.
- The database of known people must be continually cleaned up - when a person has not been seen for some period of time, they have most likely left the area and should be removed from the database.

There are two fundamentally different approaches to re-identification: Single-shot and multi-shot. Single-shot performs the re-identification on stand-alone frames. This is useful in situations where only a single probe

image is available. However, very often the subject has been captured on video, and thus has several frames describing her. Multi-shot combines a full pass across the field of view into a single model, which is the used as probe in a gallery of similarly collected multi-shot models. Multi-shot gives the option of capturing more information about the subject than a single frame contains, and has the potential to make the system more robust to occlusions and sudden changes in lighting.

Person re-identification has been in active research for a while, but multi-modal systems have only recently come into play. The reason for this is twofold: 1) Algorithms have so far mostly been developed for use in existing surveillance infrastructure and 2) more advanced sensor capabilities, such as depth and thermal, have not been readily available. We believe that as sensor technology progresses, more modalities will show up in regular surveillance cameras, making the development of new multi-modal algorithms highly relevant.

This work builds on the method presented in [8] and is a full RGB-D based re-identification system covering all parts of the pipeline from detection through re-identification to database maintenance. The main contributions are:

- While the earlier work was single-shot based, the method has been updated to a multi-shot approach. This work compares several different multi-shot person models.
- The earlier work relied on RGB-color histograms. This work presents a comparison of three different color spaces: RGB, HSV, and XYZ.
- More thorough testing. On top of testing on the original dataset from [8], two more datasets have been captured to test the performance in different circumstances.
- The system is now free of arbitrary thresholds in the re-identification stage, as every threshold is learned from training data in a cross-validation scheme.
- In the original work, the height of subjects only had little influence on the re-id performance. We introduce a more thorough pruning step based on depth-adjusted height of subjects which increases re-id performance significantly.

The remainder of this paper is structured as follows: Section 2 gives an overview of related work in the field of re-identification. It also contains a description of existing datasets, as well as the ones captured and used in this work. Section 3 explains the algorithms used and goes through detection and segmentation, multi-shot person modeling, and re-identification. In section

4 the person database is explained, and in section 5 the various methods presented are evaluated against each other. Section 6 concludes the paper.

2 Related work

Person re-identification as described above has been an active research area for about a decade and truly gained speed in the latter half of the 2000s. A relatively recent survey on person re-identification can be found in [6], and in this section we highlight notable recent papers. As mentioned previously, re-identification approaches can be divided into single-shot and multi-shot. Furthermore, we distinguish whether multi-modal methods are used.

Zheng et. al. [12] and Zhao et. al. [11] both use single shot algorithms. The first use color and texture histograms, whereas the latter uses dense color histograms and SIFT descriptors with the addition of using a saliency map to decide which parts of the person are the most descriptive.

Multi-shot is championed by Bak et. al. in [1] and Demirkus et. al. [5]. Bak uses a large pool of features and the best one to describe a particular person is selected. Demirkus uses a set of more directly understandable soft biometrics, such as gender, hair color, and clothing color.

Moving away from the traditional visible light modality, Jüngling and Arens [7], presents a full single-shot re-identification pipeline based on infrared images. It detects candidates, then tracks and re-identifies them using SIFT-features. In the depth modality, Barbosa et. al. [2] re-identifies by comparing various physical body measurements (anthropometrics) obtained from the depth image. Velardo and Dugelay [10] uses manually measured anthropometrics to prune the set of candidates for face recognition.

Finally, two papers combine several modalities. In [8] RGB is used for detection and re-identification, and depth for segmentation and pruning of re-id candidates. This is the same basic approach as in this work. In [9], thermal images and anthropometric measurements are added and the re-identification is performed in a truly multi-modal way with a combination of color histograms, SIFT features on thermal images, and anthropometric measurements obtained from depth images.

2.1 Datasets

Several public datasets exist, though mostly sets captured with traditional visible light sensors.

In other modalities, not many exist. For depth, the RGB-D Person Re-identification Dataset [2] is one option. It contains 79 people in 4 different scenarios: Walking slowly with outstretched arms, two instances of walking from a frontal viewpoint, and walking from a rear viewpoint.

	Novi	Basement	Hallway
Number of persons	22	35	10
Number of frames	7800	7231	4492
Contains image sequences	Yes	Yes	Yes
Available modalities	RGB, depth	RGB, depth, thermal	RGB, depth, thermal

Table C.1: Statistics on the three data sequences used in this work.

(a) **(b)** **(c)**

Fig. C.1: Example images from our own (a) Novi, (b) Basement, and (c) Hallway sequences.

For this work, we use our own dataset with a surveillance-like camera setup. We have three sequences: Novi, Basement, and Hallway. They all contain sequences of persons walking diagonally towards and past the sensor twice. Novi, which was also used in [8], contains 22 persons over 7800 frames (passes have varying lengths). Basement contains 35 persons over 7231 frames, and Hallway contains 10 persons over 4492 frames. Stats about the public as well as our own datasets can be seen in table C.1. The sequences were captured with Microsoft Kinect for Xbox. Example pictures from each sequence can be seen in fig. C.1.

3 Algorithm overview

This paper describes a full re-identification system which takes a raw RGB-D feed as input and outputs whether or not a passing person has been seen before, and if so, what the previous ID was. This is different from many other re-identification papers which most often describe a core algorithm without much focus on all the other system parts that must be in place to have an actual working system. The process requires several steps: Persons must be detected and segmented, they must be modeled, and finally re-identified. On top of the re-identification process comes the process of keeping tabs on the person database. A flowchart is shown in fig. C.2.

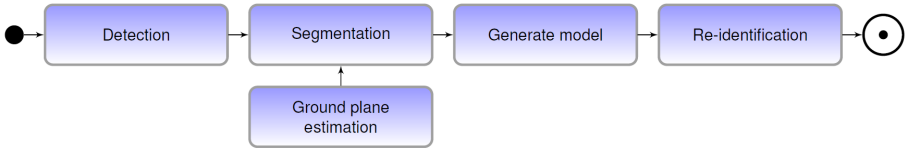


Fig. C.2: Illustration of the flow through the system.

3.1 Detection and segmentation

The detection is done with a standard HOG-detector as first proposed by Dalal and Triggs [4]. The detector is trained on the INRIA Person Dataset introduced by the same paper. The detector runs on the RGB images and returns person bounding boxes.

The detected persons need to be segmented in further detail. The bounding box is not sufficient, since we do not want to capture features from the background. Segmentation is achieved with a flood fill in the depth image. Persons not crawling on the floor are conveniently separated from the background in the depth modality, so a flood fill to similar pixels starting at the points

$$\mathbf{X} = \begin{bmatrix} 2/5 & 1/4 \\ 2/5 & 1/3 \\ 2/5 & 2/5 \\ 1/2 & 1/4 \\ 1/2 & 1/3 \\ 1/2 & 2/5 \\ 3/5 & 1/4 \\ 3/5 & 1/3 \\ 3/5 & 2/5 \end{bmatrix} \begin{bmatrix} b_w & 0 \\ 0 & b_h \end{bmatrix} + \begin{bmatrix} b_x & b_y \\ \vdots & \vdots \\ b_x & b_y \end{bmatrix}_{9x2} \tag{C.1}$$

where $\mathbf{X}$ is a 9x2 matrix containing the x and y coordinates of the flood fill points, b is the bounding box with subscript x, y, w, and h meaning top-left x-coordinate, top-left y-coordinate, width, and height respectively. The flood fill is performed at multiple positions to ensure that we have a stable object in the depth modality. A person is classified as stable if at least four of the depth points converge, i.e. the flood fill of these points fill out the same volume.

Ground plane estimation

One problem with the flood fill is that at the feet of the subject, the fill is likely to spill onto the floor. To counter this, ground plane pixels on the depth image are removed. When the system is started initially, a ground plane is defined in the depth image. This is done by marking a number of points on the ground and performing a least squares solution of the bivariate polynomial:

$$z_{\text{poly}} = a_{00} + a_{01}x + a_{02}x^2 + a_{10}y + a_{20}y^2 + a_{11}xy \tag{C.2}$$

Fig. C.3: The left image illustrates a detection. On the right, the person has been segmented in the depth image, and the blue boxes illustrates the boxes which are used as basis for the color histograms.

Although the floor is planar, the measurements of the floor from the Kinect depth sensor are representing the plane as a hyperbolic plane, thus stating the need for a bivariate polynomial. When the coefficients are determined, any pixel in the depth image close to the ground plane is colored black. Those pixels are the ones fulfilling the inequality in equation (C.3), where p is the pixel in question and t_{depth} defines the distance from the theoretical ground plane that is still considered part of that plane.

$$|z_{\text{poly}} - p_z| < t_{\text{depth}} \tag{C.3}$$

3.2 Person model

One of the objectives of this paper is to compare two types of multi-shot person models. They are both based on the two-part color histogram used in [8]: After a person is segmented, a color histogram is computed for the upper part of the body and the lower part of the body (as illustrated by the blue boxes in fig. C.3). Each color channel is divided into 20 bins, the individual channel histograms are concatenated, and finally the two part histograms are concatenated for a feature vector of $20 \cdot 3 \cdot 2 = 120$ dimensions in the case of a 3 channel color space. In addition to the two modeling paradigms, 3 different color spaces were tested: RGB, HSV, and XYZ. For HSV and XYZ the luminance channels were removed to enhance lighting invariance, so in those cases the final histogram would be 80-dimensional and contain just the HS- and XZ-channels, respectively.

Two multi-shot schemes have been tested:

1. Mean histogram of all frames in a pass.
2. All frame-histograms saved individually.

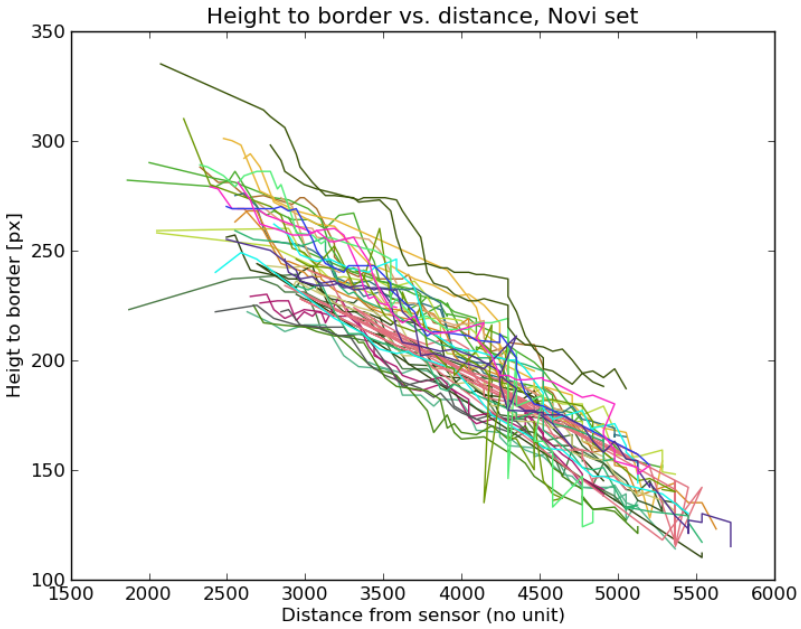


Fig. C.4: Curves depicting height-to-border versus distance for all tracks in a sequence. The curves are colored in pairs, such that two tracks of the same color are two passes by the same person. It can be seen that most lines are close to their partner of the same color, showing that the height measurement is stable across passes.

In 1) the mean histogram is computed when a pass is over. Each bin is simply averaged:

$$m_i = \frac{1}{n} \sum_{j=0}^{n} h_{i,j} \text{ for } 0 \leq i < k \tag{C.4}$$

where m is the mean histogram, n is the number of frames in the pass, k is the number of bins in the histograms and $h_{i,j}$ is the value of bin i in histogram j.

In 2) no averaging takes place. Instead a pass is modeled after each histogram in it. See the following section on how each model is matched against the person database.

Both of the color-based models are augmented with a measure of the person's height. We use normalized height-to-border. This is the distance in pixels from the top of the person in the image, to the bottom of the frame, normalized by the depth of the observation. This reduces noise, as only one of the bounds of the height is now determined from the noisy depth sensor. It also allows for clipping.

In fig. C.4 height-to-border versus depth is plotted. Because the surface

and field-of-view is the same for all who pass by the camera, the only change that will happen to the curve for people of different heights is a shift in its y-axis intercept. Instead of approximating the full curve, we go for the less computationally heavy option of modelling each pass with the mean of the depth-normalized height-to-border, designated γ, for all instances in the pass:

$$\gamma = \frac{1}{n} \sum_{i=0}^{n} g_i \cdot d_i \tag{C.5}$$

where g_i is the height-to-border for observation i in the pass, and d_i is the distance to the person in that observation. While the person is not completely flat, for the purpose of this normalization, we use the depth of the seed point described in equation C.1.

3.3 Re-identification

A pruning stage based on the height measurement is used before the re-identification. The height of the probe is compared to the gallery by means of the absolute difference in their heights. If the mean normalized height-to-border is more than t_h away from a candidate, the candidate is not considered a match for this subject. t_h is found from analyzing training data before running the system. The threshold t_h is set to the mean of the height difference between wrong matches in the training set.

When re-identifying, the model of the current pass is compared to those of the persons in the database, which is initially empty, but will be built as time progresses. Both the mean histogram and the histogram series model use the Bhattachariyya distance [3]:

$$d(H_1, H_2) = \sqrt{1 - \sum_I \frac{\sqrt{H_1(I) H_2(I)}}{\sqrt{\sum_I H_1(I) \cdot \sum_I H_2(I)}}} \tag{C.6}$$

where $d(H_1, H_2)$ is the distance between the histograms H_1 and H_2, and $H(I)$ is the value of bin I in the histogram H. The result is a number between 0 and 1, where 0 is a perfect match.

With mean histograms, where only two histograms - probe and gallery - are involved, the distance itself is used, and the subject is either re-identified, ignored, or added to the database. With histogram series, the model comprise a series of histograms. In this case, each histogram in the probe model is compared to each histogram in the database. The probe then casts a vote for the ID of the gallery-model which contains the histogram it is closest to, if that is within a separately trained ignore threshold. The gallery-model with the most votes is selected as the best candidate, provided is has the majority (more than 50%) of the possible votes.

3.4 Mean histogram

The re-identification process is governed by two thresholds:

$$t_n\text{: New threshold: Subjects with } d(H_1, H_2) > t_n \text{ are added as new persons} \tag{C.7}$$

$$t_i\text{: Ignore threshold: Subjects with } d(H_1, H_2) <= t_i \text{ are re-identified} \tag{C.8}$$

This implicates that subjects with $t_i < d(H_1, H_2) <= t_n$ are ignored, because they are too similar to other subjects, without being similar enough to trust the identification.

The thresholds are learned beforehand by observing a training set. The distances between all mean histograms in the training set are computed and stored in the set $\mathcal{D}$ and divided into two sets $\mathcal{D}^c$ and $\mathcal{D}^w$ where $\mathcal{D}^c$ contains distances between different observations of the same person and $\mathcal{D}^w$ contains distances between histograms of different persons:

$$\mathcal{D}^c = \{\mathcal{D} | id(H_1) = id(H_2) \text{ in } d(H_1, H_2)\} \tag{C.9}$$

$$\mathcal{D}^w = \{\mathcal{D} | id(H_1) \neq id(H_2) \text{ in } d(H_1, H_2)\} \tag{C.10}$$

where $id(\bullet)$ is the person id connected with a histogram. The thresholds are then computed as:

$$t_n = \overline{\mathcal{D}^w} - 2 \cdot \sigma(\mathcal{D}^w) \tag{C.11}$$

$$t_i = \overline{\mathcal{D}^c} + \sigma(\mathcal{D}^c) \tag{C.12}$$

where $\overline{\bullet}$ denotes mean and $\sigma(\bullet)$ denotes standard deviation.

3.5 Histogram series

The re-identification for the histogram series model uses many of the same principles of the mean histogram model, but is adapted to use many more histograms for each subject to encompass variations in lighting and pose. A histogram is computed for each frame in the pass of a subject and they are then compared to all histograms already in the database. When the shortest distance d_s to any gallery-histogram is less than t_i, the associated person id, p_s receives a vote. Thus, each subject histogram contributes with up to 1 vote, for a theoretical total of $len(\mathbf{H})$ votes: the number of histograms in the current pass. If there are no histograms in the pass, the subject is ignored. If any person in the gallery has received more than half the theoretical maximum, the subject is re-identified as him. If no gallery person satisfies this requirement, the subject is added as a new person.

It is worth noting that this method has no explicit option of ignoring the subject in case it is uncertain, other than in the case where no histograms exist.

4 Transient database

A crucial part of real-world re-identification is that the gallery database in not known beforehand, but must be built and updated continuously. Furthermore, in a short-term re-identification system, it can be expected that a person who has not been seen for some amount of time has left the area the system is concerned with and should be removed.

5 Evaluation

6 permutations of the system have been tested on 3 different sequences (see section 2.1). The 2 different multi-shot models have both been tested in 3 different color spaces: RGB, HSV, and XYZ. To counter variations in lighting, only HS and XZ were used for HSV and XYZ.

The performance of the system varies with the order the persons are passing by the camera. If a person that is very hard to re-identify passes by the camera in the first two passes without any other entries in the database, odds are that he will be correctly re-identified. However, if a similar person enters the database before the second pass of person 1, they might be confused with each other and thus lower the performance. To even out this effect, all results presented below are averages of 100 runs where the subjects enters the system in random order. That should sufficiently even out any "lucky" or "unlucky" orderings and provide accurate results. For each run, all thresholds have been trained on a random subset of 20% of the sequence, which is then excluded from the rest of the run. The effect of the training set selection should also average out.

The re-identification performance can be characterized with 5 parameters:

1. Correct new
2. Wrong new
3. Correct ID
4. Wrong ID
5. Ignored

The first two describes how well the system distinguishes between known persons and new persons. Ideally, there should be no wrong new, as they are persons that are already in the database and should have been re-identified. Correct ID and wrong ID comprises the subjects that are neither ignored, correct new, nor wrong new, but are re-identified. Finally, ignored are the ones that are not handled because they are neither close enough to an existing

	Basement seq.	Hallway seq.	Novi seq.
Mean observation length:	11.8	4.0	23.1
Median observation length:	12	3	23
Minimum observation length:	0	0	5
Maximum observation length:	22	10	40

Table C.2: Statistics on the amount of observations of captured persons for each sequence. The numbers are based on the amount of times a single person was detected and modeled in a single pass.

person to be re-identified, nor different enough from the existing persons to be added to the database.

The results of the tests can be seen in table C.3. Sequence length and detection performance varies greatly between sequences, as seen in table C.2. Especially the Hallway sequence contains many shorter tracks, meaning that generalization, as well as the benefit from the multi-shot approach, declines heavily.

Generally, the mean histogram approach performs better than histogram series. The histogram series has comparable or better re-id performance percentage-wise in some cases, but in absolute numbers, the performance is worse, with significantly more wrong new and significantly lower re-id numbers. The number of wrong identifications is low across the board, so the weak spots are the wrong new- and ignored-counts which are rather high. Most new passes are correctly classified as such, at around 30 of 35 in the basement sequence, 8/10 and 21/22 in the Hallway and Novi sequences respectively.

The benefit of the ignore-functionality in the mean histogram model is illustrated in fig. C.5. Blue columns are a histogram of distances between mean histograms of the same person, while red columns are a histogram of distances between different persons. The overlap between these shows that is it not possible to achieve perfect classification with a 1d decision boundary in this case. To counter this, an ignore zone is introduced - the space between the green and the yellow line, the thresholds, which can to some extent mitigate the effects of this overlap. In reality, when training on a subset of the data, the ignore zones are generally wider than in this example. It is possible that a classification in a higher dimensional space would work better and allow discarding the ignore zone.

Table C.4 shows how the height-based pruning step improves the re-id rates across all methods. By discarding obviously wrong candidates based on height, the correct re-id rate goes up by 4.53 percentage points on average.

We have been unable to compare our results to the work of others, as they do not present full-flow systems, but rely on tightly pre-cropped images of

Basement sequence

		Correct new	Wrong new	Correct ID	Wrong ID	Ignored	% correct	% wrong
RGB	Mean histogram	29.31 (2.92)	4.53 (7.10)	11.76 (5.34)	1.22 (2.50)	8.18 (6.82)	90.62 %	9.38 %
	Histogram series	32.61 (1.44)	8.64 (7.37)	13.00 (7.07)	0.74 (2.14)	**0.00** (0.00)	94.60 %	5.40 %
HS	Mean histogram	28.75 (3.23)	**3.20** (7.36)	12.57 (5.81)	1.18 (2.83)	9.30 (6.78)	91.43 %	8.57 %
	Histogram series	**32.71** (1.37)	8.13 (7.77)	**13.49** (7.51)	**0.67** (2.27)	**0.00** (0.00)	**95.24** %	**4.76** %
XY	Mean histogram	29.02 (3.26)	4.72 (7.12)	11.24 (5.15)	1.68 (3.01)	8.34 (7.35)	86.97 %	13.03 %
	Histogram series	32.54 (1.45)	8.88 (7.34)	12.59 (6.87)	0.98 (2.33)	**0.00** (0.00)	92.78 %	7.22 %

Hallway sequence

		Correct new	Wrong new	Correct ID	Wrong ID	Ignored	% correct	% wrong
RGB	Mean histogram	8.36 (1.00)	4.07 (2.34)	1.15 (1.37)	0.61 (1.01)	0.81 (1.62)	65.34 %	34.66 %
	Histogram series	8.59 (0.77)	4.45 (2.10)	1.30 (1.57)	0.66 (1.18)	**0.00** (0.00)	66.33 %	33.67 %
HS	Mean histogram	8.15 (1.17)	**3.95** (2.41)	1.34 (1.61)	**0.39** (0.78)	1.17 (2.13)	**77.46** %	**22.54** %
	Histogram series	**8.61** (0.62)	4.44 (2.14)	**1.44** (1.72)	0.51 (0.76)	**0.00** (0.00)	73.85 %	26.15 %
XY	Mean histogram	8.37 (0.96)	4.11 (2.29)	1.15 (1.37)	0.63 (1.10)	0.74 (1.58)	64.61 %	35.39 %
	Histogram series	8.57 (0.71)	4.33 (2.14)	1.37 (1.58)	0.73 (1.22)	**0.00** (0.00)	65.24 %	34.76 %

Novi sequence

		Correct new	Wrong new	Correct ID	Wrong ID	Ignored	% correct	% wrong
RGB	Mean histogram	21.15 (1.40)	3.79 (2.98)	9.68 (2.73)	0.25 (1.31)	1.13 (1.89)	97.48 %	2.52 %
	Histogram series	**21.51** (0.70)	9.12 (4.21)	5.31 (4.24)	**0.06** (0.42)	**0.00** (0.00)	**98.88** %	**1.12** %
HS	Mean histogram	20.64 (1.86)	**2.42** (2.13)	**10.52** (2.38)	0.44 (1.45)	1.98 (3.21)	95.99 %	4.01 %
	Histogram series	21.48 (0.77)	9.18 (4.53)	5.23 (4.59)	0.11 (0.91)	**0.00** (0.00)	97.94 %	2.06 %
XY	Mean histogram	21.14 (1.17)	4.89 (3.48)	8.83 (3.10)	0.34 (1.26)	0.80 (1.74)	96.29 %	3.71 %
	Histogram series	21.46 (0.87)	10.12 (3.91)	4.31 (3.93)	0.11 (0.91)	**0.00** (0.00)	97.51 %	2.49 %

Table C.3: Re-identification performance of the 6 system configurations on 3 different sequences. All numbers are averaged over 100 runs with random enrollment order.

		Without height		With height		Difference	
		% correct	% wrong	% correct	% wrong	% correct	% wrong
Basement	Mean histogram	82.17 %	17.83 %	90.67 %	9.33 %	8.50 %	-8.50 %
	Histogram series	87.28 %	12.72 %	94.21 %	5.79 %	6.93 %	-6.93 %
Hallway	Mean histogram	64.64 %	35.36 %	69.14 %	30.86 %	4.50 %	-4.50 %
	Histogram series	67.34 %	32.66 %	68.47 %	31.53 %	1.10 %	-1.10 %
Novi	Mean histogram	92.03 %	7.97 %	96.59 %	3.41 %	4.56 %	-4.56 %
	Histogram series	96.50 %	3.51 %	98.11 %	1.89 %	1.61 %	-1.61 %
Average		81,66 %	18,34 %	86,20 %	13,80 %	**4,53 %**	-4.53 %

Table C.4: Comparison of re-identification performance with and without the height-based candidate pruning step.

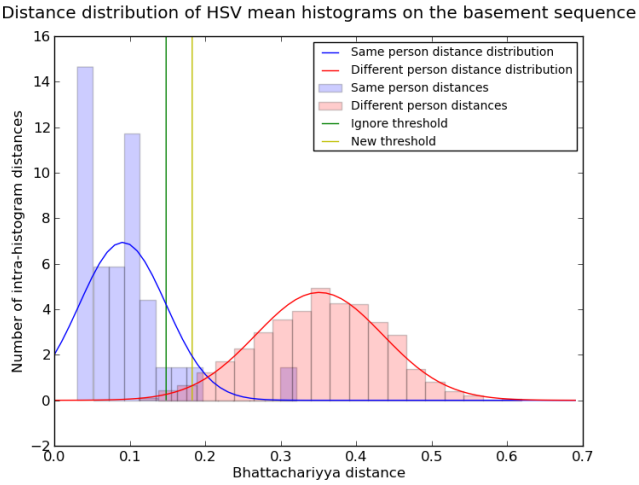


Fig. C.5: Distribution of distances between histograms in the full basement sequence. There is a clear overlap of distances between histograms from the same person and histograms from different persons. When using a distance threshold to classify, this will result in wrong identifications. The ignore-threshold allows to remove the distances that are the most affected by this overlap.

persons. Furthermore, our system needs depth images as well as RGB, so no existing dataset has been compatible. We also do not present CMC-curves as that ranking system works poorly for on-the-fly enrollment systems, where, in many cases, there are simply not enough entries in the database to do a proper ranking.

We can, however, compare some of our results to the work previously presented in [8]. Not all stats are directly comparable, but the correct and wrong ID rates are. In that work, they are 68% and 0%, with an ignore rate of 24%. The system presented here has a much higher correct ID rate, but at the cost of a somewhat higher wrong ID rate.

6 Conclusion

This work presented a re-identification system using RGB-D data and compared several model and color space configurations. It introduces 3 new, different re-identification sequences for testing, and goes through all stages from candidate detection to identification. Furthermore, it investigates how to handle online enrollment of subjects, a subject few previous works have touched. Future work includes more sophisticated multi-shot models, and enhancing the system to cope with multiple, co-occluding subjects in crowded environments.

References

[1] S. Bak, G. Charpiat, E. Corvée, F. Brémond, and M. Thonnat, "Learning to Match Appearances by Correlations in a Covariance Metric Space," in *ECCV (3)*, ser. LNCS, vol. 7574. Springer, 2012, pp. 806–820.

[2] I. B. Barbosa, M. Cristani, A. D. Bue, L. Bazzani, and V. Murino, "Re-identification with RGB-D Sensors." in *ECCV Workshops (1)*, ser. LNCS, vol. 7583. Springer, 2012, pp. 433–442.

[3] G. Bradski and A. Kaehler, *Learning OpenCV*. O'Reilly, 2008, ch. 7, pp. 201–202.

[4] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *CVPR*, 2005.

[5] M. Demirkus, K. Garg, and S. Guler, "Automated person categorization for video surveillance using soft biometrics," pp. 76 670P–76 670P–12, 2010.

[6] G. Doretto, T. Sebastian, P. H. Tu, and J. Rittscher, "Appearance-based person reidentification in camera networks: problem overview and current approaches," *J. Ambient Intelligence and Humanized Computing*, vol. 2, no. 2, pp. 127–151, 2011.

[7] K. Jüngling and M. Arens, "Local Feature Based Person Reidentification in Infrared Image Sequences." in *AVSS*. IEEE Computer Society, 2010, pp. 448–455.

[8] A. Møgelmose, T. B. Moeslund, and K. Nasrollahi, "Multimodal Person Re-Identification using RGB-D Sensors and a Transient Identification Database," in *International Workshop on Biometrics and Forensics*, 2013.

[9] A. Møgelmose, A. Clapés, C. Bahnsen, T. B. Moeslund, and S. Escalera, "Tri-modal Person Re-identification with RGB, Depth and Thermal Features," in *9th IEEE Workshop on Perception Beyond the Visible Spectrum*. IEEE, 2013.

[10] C. Velardo and J. Dugelay, "Improving Identification by Pruning: A Case Study on Face Recognition and Body Soft Biometric," in *WIAMIS*. IEEE, 2012, pp. 1–4.

[11] R. Zhao, W. Ouyang, and X. Wang, "Unsupervised Salience Learning for Person Re-identification." CVPR, 2013.

[12] W. Zheng, S. Gong, and T. Xiang, "Person re-identification by probabilistic relative distance comparison," in *CVPR*. IEEE, 2011, pp. 649–656.

Paper D

The AAU Multimodal Annotation Toolboxes: Annotating Objects in Images and Videos

Chris H. Bahnsen, Andreas Møgelmose, and Thomas B. Moeslund

This technical report has been published at
arXiv.org, 2018.

The layout has been revised.

Abstract

This tech report gives an introduction to two annotation toolboxes that enable the creation of pixel and polygon-based masks as well as bounding boxes around objects of interest. Both toolboxes support the annotation of sequential images in the RGB and thermal modalities. Each annotated object is assigned a classification tag, a unique ID, and one or more optional meta data tags. The toolboxes are written in C++ with the OpenCV and Qt libraries and are operated by using the visual interface and the extensive range of keyboard shortcuts. Pre-built binaries are available for Windows and MacOS and the tools can be built from source under Linux as well. So far, tens of thousands of frames have been annotated using the toolboxes.

1 Introduction

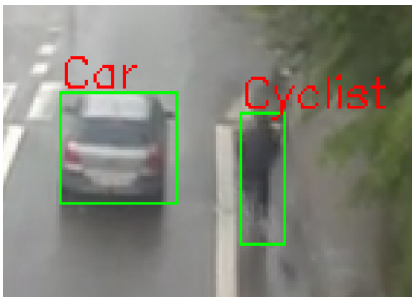
The main driver behind modern computer vision systems is annotated data - and lots of if. If one wants to train, test, benchmark or verify any vision algorithm that addresses a real-world problem, you need real-world annotated data. You might be lucky that a suitable dataset for your problem exists but often you will need new annotated data that suits your domain. For many years, this has been the case for most of our work at the Visual Analysis of People Laboratory at Aalborg University. Through a collaborative effort at our lab, we have created two separate annotation tools that can be compiled to run under Windows, MacOS, and Linux.

The *AAU VAP Multimodal Pixel Annotator* may be used to annotate pixel-based masks of object instances whereas the *AAU VAP Bounding Box Annotator* may be used to annotate bounding boxes around objects of interest. Both annotation tools support annotation tags such that an annotated object may be associated with a predefined class name. Example annotations, both pixel-based and bounding box-based, are shown in Figure D.1.

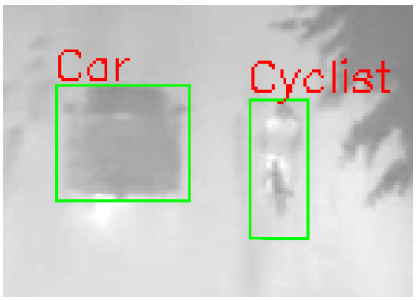
In this text, we will give an overview of the two annotation tools and the features they provide. An updated list of all annotation tools offered by our laboratory is found at Bitbucket[1]. The source code and binaries of the two annotation tools are available under the MIT license.

The annotation tools have been used to annotate humans [2, 14], road users [1], road signs [9], chicken entrails [11], pigs, fish [6], material defects, and more. The number of annotated frames in the examples above vary from a few hundred to tens of thousands. In the next section, we will describe the common features of the two annotation tools. Section 3 describes the specific features of the Bounding Box Annotator whereas Section 4 gives a description of the Multimodal Pixel Annotator. Section 5 concludes the work so far and gives insights on the future development of the toolboxes.

[1] `https://bitbucket.org/account/user/aauvap/projects/AN`

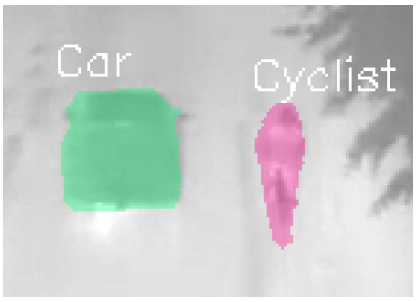


(a) Bounding box annotation in RGB



(b) Corresponding bounding box annotation in thermal

(c) Pixel annotation in RGB



(d) Corresponding pixel annotation in thermal

Fig. D.1: Bounding box and pixel-based samples of the same objects annotated in both RGB and thermal modalities. Every annotation is associated with a corresponding tag.

2 Common Features

The annotation tools are developed in C++ with Qt and OpenCV [3] as the main libraries. Both tools have been developed in parallel and thus share many features and much of the code base. The shared features are described below.

2.1 Object Properties

Every annotated object is associated with a unique identification number (ID), a class tag, and optionally one or more meta data tags. An example hereof is shown in Figure D.2.

We will go through the object properties below. Properties shown in **bold** are mandatory whereas properties shown in *italics* are optional.

- **Tag** The class name of an object. The class name may be freely chosen or

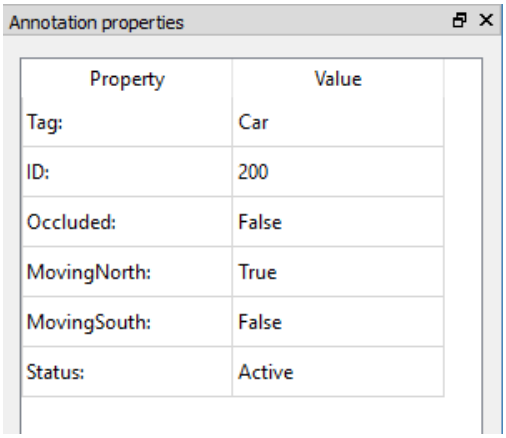


Fig. D.2: Object properties of an annotation. The "Occluded", "Moving North", and "Moving South" entries are meta data tags that may be either true or false.

limited to a pre-defined list if the setting `Limit annotation tags to suggested list` is checked. The suggested list is populated from the existing annotation tags in the dataset and from the user-editable list available in `File` → `Edit suggested tags`.

- **ID** The identification number of the object. In Bounding Box Annotator, this number is defined in the range $[0, \text{inf}]$ and is unique for the entire annotation sequence. In Multimodal Pixel Annotator, the ID is encoded into the mask image which limits the range to the interval from $[0, 255]$. However, the ID's in the range from [0,10] are reserved for internal operations of the program whereas ID 170 is reserved for don't care borders.

- *Meta data tags* The meta data tags are binary object attributes. The meta data names themselves may be specified before creating an annotation sequence in `File` → `Edit meta data fields` or retrospectively applied by manually editing the csv-file containing the annotations. Three meta data names have been set in Figure D.2: the "Occluded", "Moving North", and "Moving South" tags. These tags may be either true or false for an object and are defined for every frame.

- **Status** When annotating video data as described in Section 2.2, one might choose to copy existing annotations to temporally adjacent frames. However, an object might be moving out of the image frame and as a result, the annotated mask belonging to this object should not be copied to the next frame. This might be changed by setting the object status from `Active` to `Last frame reached`.

Fig. D.3: Buttons from left to right: (1) Retain image when loading previous frame, (2) Retain image when loading next frame, (3) Interpolate between annotations when stepping > 1 frames.

2.2 Annotation of Sequential Data

The annotation toolboxes assume that the source images are in the same folder. The toolboxes do not directly support video files, mainly because OpenCV does not provide efficient and accurate temporal search for videos. Instead, videos may be converted to a collection of single frames by an FFMPEG script[2]. One may configure the annotation toolboxes such that they only load frames that adhere to a specific file pattern. The option is set in `Settings` $\rightarrow$ `File patterns` and supports regular expressions. For simple use cases such as including all .png-files, the string `*.png` is sufficient.

Retaining annotations in adjacent frames When annotating frames that are temporally consistent, i.e. the same objects are moving slowly from frame to frame, it might be useful to copy the annotations from frame n to frame $n+1$ or $n-1$. This functionality is found in the `Retain when loading previous` and `Retain when loading next` buttons illustrated in Figure D.3.

2.3 Multi-Modal Annotation

Both annotation tools support the annotation of objects in two views and given the preference in our lab for multi-modal approaches [8], we refer to view 1 as RGB and view 2 as thermal. The RGB modality is the master modality and all annotations are by default stored in a coordinate system relative to the RGB image coordinates. For compatibility with the AAU Trimodal People Segmentation Dataset[3], the Multimodal Pixel Annotator also enables a depth modality which is currently in legacy support.

Registration from RGB $\leftrightarrow$ Thermal can be performed using a single homography which may be sufficient if the objects of interest in the scene are lying in close proximity to the same plane. The homographies should be stored in a yml-file using the OpenCV FileStorage method in the `homRgbToT` and `homTToRgb` variables. Example homographies are found from the sample annotations provided at the Bitbucket project pages.

If the planar constraint is violated and a single homography is not sufficiently accurate, one may use a combination of multiple homographies. More details about this approach are found in the work by Palmero et al. [10].

[2] `ffmpeg -i file.mpg -r 1/1 %05d.png`

[3] `https://www.kaggle.com/aalborguniversity/trimodal-people-segmentation`

Fig. D.4: The don't care mask of the image is overlaid in yellow. The colour and opacity of the mask may be defined by the user.

2.4 Don't Care Masks

It might be beneficial to use a don't care mask that visualizes the region-of-interest in which objects should be annotated. If this option is enabled in settings, a binary mask image should be placed in the root folder of the annotations or the directory above. If the don't care mask is placed here under the name `mask.png`, the mask will be loaded automatically when opening an annotated sequence. An example of a don't care mask is shown in Figure D.4.

2.5 Shortcut-driven Annotations

Maximizing the use of the keyboard is one of the better ways of speeding up the annotation process. Besides the mouse-driven drawing functionality, almost every other aspect of the annotation tools may be operated by using the keyboard. The respective shortcuts are revealed by hovering the mouse on top of each button. Alternatively, the wiki pages[4,5] of the annotation tools provide a great overview of the available shortcuts.

3 Bounding Box Annotator

The Bounding Box Annotator provides an interface for drawing bounding boxes around objects of interest. It provides additional features for working

[4] `https://bitbucket.org/aauvap/bounding-box-annotator/wiki/Home`
[5] `https://bitbucket.org/aauvap/multimodal-pixel-annotator/wiki/Home`

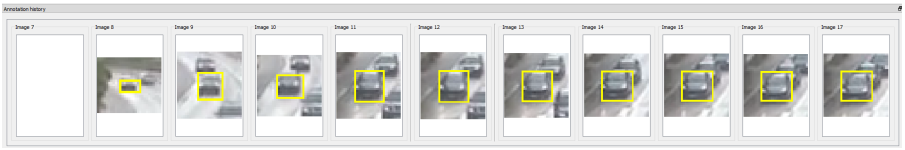


Fig. D.5: The annotation history window of the Bounding Box Annotator. The selected annotation of the current frame (Image 12) is shown in the middle, surrounded by annotations containing the same ID in the previous and next five frames. Image 7 is empty, indicating that the object ID does not exist in this frame.

with image sequences such as interpolation and extended annotation deletion and merging functionality.

3.1 Temporal Interpolation

When working with image sequences with high frame-rate and slow-moving objects, annotating every single frame is usually a very tedious task. The Bounding Box Annotator attempts to ease the annotation process by:

- Providing an overview of annotations with the same ID in the neighbouring frames, illustrated in Figure D.5.
- Interpolating between annotations. If the user annotates an object in frame 1 and frame 6, the program optionally interpolates between these annotations to create corresponding annotations for frame 2, 3, 4, and 5. Best results are achieved when the motion of the object is nearly linear.

3.2 Deleting and Merging Annotations

When using the 'retain image' buttons illustrated in Figure D.3, one might forget to set the `Last frame reached` flag, leading to several duplicate annotations once the object of interest has left the frame. The button `Delete selected annotations in current and future frames` comes to the rescue, effectively deleting annotations with the selected ID(s) in all future annotations. The program will inform the user about the affected annotations, hopefully minimizing the risk of deleting a bunch of annotations by accident. A sample prompt is shown in Figure D.6.

Two annotations might be merged by using the `Merge selected annotation and another annotation in current and future frames` button, which will do just that. After merging, the original 'other' annotation will be deleted as described in Figure D.7.

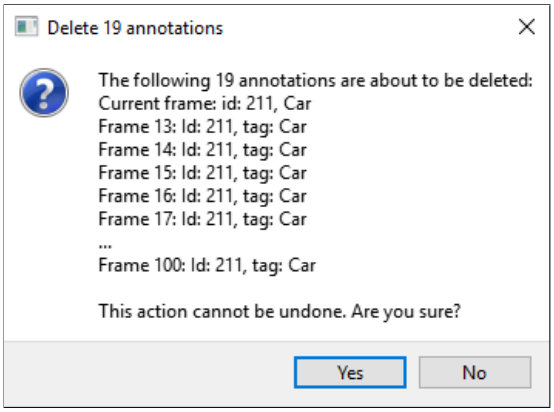


Fig. D.6: Deleting annotations with ID 211 in the current and subsequent frames. The user is asked to acknowledge the severity of this action before deletion.

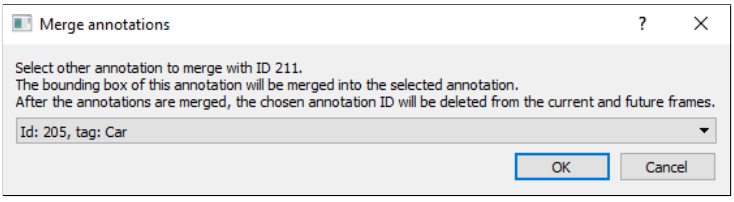


Fig. D.7: Merging an annotation ID with the currently selected annotation ID in the current and subsequent frames.

3.3 Automatic Backup

The .csv-file containing the annotations is automatically copied to a backup folder whenever an annotation folder is opened with the Bounding Box Annotator. The backup file is timestamped such that the user may easily revert to an older revision if the current annotations are deleted by accident.

3.4 Exporting Annotations

The Bounding Box Annotator saves the annotations in a single file, by default named `annotations.csv`. Each annotated object represents a line in the csv-file and the bounding box is encoded by saving the pixel coordinates of the upper left corner and the lower right corner. However, it is unlikely that this is the format of your favourite machine learning algorithm.

Currently, the Bounding Box Annotator is capable of exporting the annotations to the format used by the YOLO network running on Darknet [12]. When training a network on Darknet, every image should have a corresponding annotation file where each line indicates the category ID, centre point (X,Y), width, and height of an annotated object, all in normalized image coordinates[6]. The tag of an annotated object is translated to the corresponding category ID by selecting an appropriate category list. Out of the box, the tool comes with category lists for MSCOCO [7], ImageNet-1000 [4], YOLO-9000 [12], and PASCAL VOC [5]. If one wants to use his own list, it can be added in the `categoryLists` folder in the root directory of the program.

4 Multimodal Pixel Annotator

The Multimodal Pixel Annotator allows fine-grained pixel-level annotations. The specific functionality of the annotation tools is described below.

4.1 Drawing the mask

The user has three options for drawing a mask using the pixel annotation tool:

1. Initializing the mask and refining it using GrabCut [13].
2. Using paint-style brush tools.
3. Defining a contour around the object of interest using the polygon tool.

The graphical buttons for drawing the mask are shown in Figure D.8.

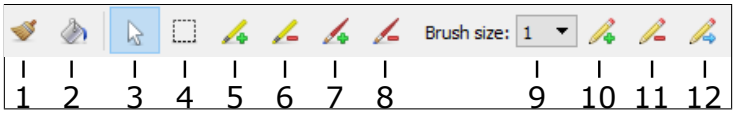


Fig. D.8: Drawing tools in Multimodal Pixel Annotator. The numbers refer to the following:
1) Removing noise from the mask.
2) Filling holes in the mask.
3) Selecting an annotation.
4) Initializing GrabCut.
5-6) Adding true positive/negative brushes to the GrabCut mask.
7-8) Manually add to/remove from mask.
9) Define brush size of tools 5-8.
10-12) Add/remove/move point from polygon mask.

(a) Initializing GrabCut **(b)** Adding true positives (red) **(c)** The resulting GrabCut mask

Fig. D.9: Example use of the GrabCut tools. Steps b)-c) are performed iteratively until the mask covers the object of interest.

Using GrabCut

When using GrabCut, the user should initialize a bounding box around the object of interest. If the appearance of the object is significantly different from the background, chance is that the initial GrabCut segmentation may be good enough. If that is not the case, the user may supply ground truth positive and negative brushes to guide the GrabCut segmentation. An example is shown in Figure D.9. Please keep in mind that GrabCut segmentation is an iterative process and the entire mask may change whenever true positive and negative brushes are drawn. If one wants to apply final touches to an otherwise finished mask, the manual brush tools should be used.

[6]Curiously, the output format of YOLO/Darknet is not the same as the input format.

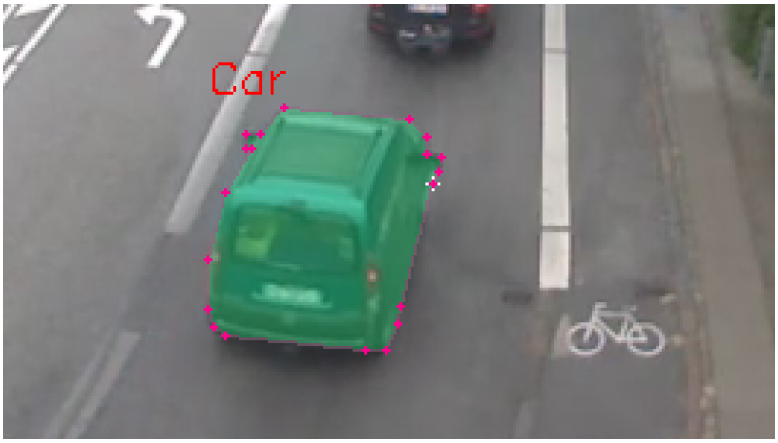


Fig. D.10: Drawing a polygon around the annotated object.

Manually Painting the Mask

If the segmentation results of the GrabCut approach are not satisfactory, the manual brush tools may be used instead. A variety of different brush sizes are provided to fit the size of the object of interest.

Drawing Polygons

If the objects to be annotated are rigid, with well-defined borders and without holes, it might be beneficial to draw the points defining the outer contour of the object. This is made possible by using the polygon tools and placing points around the outline of the object. A sample annotation using the polygon-based tools is shown in Figure D.10.

Don't Care Borders

To allow for ambiguous segmentation results around the border of objects, one can add a *don't care border* around the object masks. This option is available as "annotation borders" in `File` → `Settings` → `Annotations`. The width of the don't care border is also configurable from these settings. The don't care border is encoded in the masks with grey-scale value 170.

Filtering the Mask

The annotated mask might contain unwanted noise in the form of isolated pixels or small holes in the mask. These two problems are often encountered when using the GrabCut tools and can be easily resolved using the `Remove noise` and `Fill holes` functions depicted in Figure D.8.

4.2 Exporting Annotations

The Multimodal Pixel Annotator maintains a list of the annotations in a single csv-file, with every annotated object containing one line in the annotation. If only the polygon tools are used, the file is self-contained. On the other hand, annotated masks created using the GrabCut or brush tools are saved as grey-scale images where the annotation ID determines the shade of grey of the mask. In this case, the csv-file keeps track of the image files, the tag names, and the meta data tags.

There are currently two options for exporting the annotations:

- Converting the annotations in a bounding box-format supported by the Bounding Box Annotator.
- Exporting the annotations to a format compatible with the COCO API [7]. This creates a single json-file containing a list of all annotated images, a list of object classes, and a list of annotations either represented as polygons or compressed using run-length encoding.

5 Conclusion and Future work

This concludes the brief tour of our image annotation tools. The tools have been valuable for many different purposes in our laboratory and we sincerely hope that they will be useful for future annotation projects as well. Our laboratory have annotated tens of thousands of frames using the annotation tools and it is our experience that once one gets acquainted with the work-flow and the shortcuts, these tools provide a good environment for hours, weeks, and months of annotation work. Since the annotation tools are developed as side-line projects during our PhD's, there might be some occasional rough edges when using the programs. If the reader encounters any unexpected behaviour during the use of the programs, he or she is more than welcome to open an issue on Bitbucket.

In the future, we expect to merge the code base of the two annotation programs such that a bounding box annotation is a special case of a polygon-based annotation which again is a special case of a pixel-based annotation. If resources and time allow, we might even investigate semi-supervised annotation methods that could speed up the annotation process.

Acknowledgements

We greatly appreciate the work of our student annotators during the years and the many hours that they have spent using the programs. Their continued

work has uncovered numerous bugs which is critical in developing annotation tools that work as intended.

References

[1] T. Alldieck, C. H. Bahnsen, and T. B. Moeslund, "Context-aware fusion of rgb and thermal imagery for traffic monitoring," *Sensors*, vol. 16, no. 11, p. 1947, 2016.

[2] T. Alldieck, M. Kassubeck, B. Wandt, B. Rosenhahn, and M. Magnor, "Optical flow-based 3d human motion estimation from monocular video," in *German Conference on Pattern Recognition*. Springer, 2017, pp. 347–360.

[3] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, 2009, pp. 248–255.

[5] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[6] A. Karpova and J. B. Haurum, "Re-identification of zebrafish using metric learning," *Unpublished Master Thesis, Aalborg University, Aalborg, Denmark*, 2018.

[7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[8] A. Mogelmose, C. Bahnsen, T. Moeslund, A. Clapes, and S. Escalera, "Tri-modal person re-identification with rgb, depth and thermal features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 301–307.

[9] A. Møgelmose, D. Liu, and M. M. Trivedi, "Traffic sign detection for us roads: Remaining challenges and a case for tracking," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*. IEEE, 2014, pp. 1394–1399.

[10] C. Palmero, A. Clapés, C. Bahnsen, A. Møgelmose, T. B. Moeslund, and S. Escalera, "Multi-modal rgb–depth–thermal human body segmentation," *International Journal of Computer Vision*, vol. 118, no. 2, pp. 217–239, 2016.

[11] M. P. Philipsen, J. V. Dueholm, A. Jørgensen, S. Escalera, and T. B. Moeslund, "Organ segmentation in poultry viscera using rgb-d," *Sensors*, vol. 18, no. 1, p. 117, 2018.

[12] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint*, 2017.

[13] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM transactions on graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 309–314.

[14] A. A. Sangüesa, T. B. Moeslund, C. H. Bahnsen, and R. B. Iglesias, "Identifying basketball plays from sensor data; towards a low-cost automatic extraction of advanced statistics," in *Data Mining Workshops (ICDMW), 2017 IEEE International Conference on*. IEEE, 2017, pp. 894–901.

Paper E

Context-Aware Fusion of RGB and Thermal Imagery for Traffic Monitoring

Thiemo Alldieck, Chris H. Bahnsen, and Thomas B. Moeslund

The paper has been published in
Sensors, vol. 16, nbr. 11, 2016.

Please note that the above article is updated slightly. The updates are:

- An additional reference has been included [2].
- 77 citations to this reference are added.

The additional reference is a Master's thesis: Thiemo Andreas Alldieck, *Information based multimodal Background Subtraction for Traffic Monitoring Applications*, Master's thesis, Aalborg University, Denmark, 2015.
There is an overlap between the content of that thesis and the content of the article. The citations are added to make this overlap evident.

The layout has been revised.

Abstract

In order to enable a robust 24-h monitoring of traffic under changing environmental conditions, it is beneficial to observe the traffic scene using several sensors, preferably from different modalities. To fully benefit from multi-modal sensor output, however, one must fuse the data. This paper introduces a new approach for fusing color RGB and thermal video streams by using not only the information from the videos themselves, but also the available contextual information of a scene. The contextual information is used to judge the quality of a particular modality and guides the fusion of two parallel segmentation pipelines of the RGB and thermal video streams. The potential of the proposed context-aware fusion is demonstrated by extensive tests of quantitative and qualitative characteristics on existing and novel video datasets and benchmarked against competing approaches to multi-modal fusion.

1 Introduction

In order to increase road safety or address the problems of road congestion, one must obtain a thorough understanding of road user behavior. Such an understanding may be derived from detailed, accurate information of the traffic. Video surveillance offers a rich view of a traffic scene and enables 24-h monitoring at a fairly low cost [15]. Manual observation of the traffic scene is a tedious and time-consuming task, however, and automated techniques are thus desired. Computer vision techniques enable the automatic extraction of relevant information from the surveillance video, such as the position and speed of the traffic and the classification of the road user types [3].

The use of cameras for monitoring purposes, however, introduces a significant drawback. As the functional principle of a camera builds on the visual range of light, the quality of the data is highly dependent on environmental conditions, such as rain, fog and the day and night cycle. As a result, many applications work only during the daytime in decent weather conditions, and a persistent monitoring of the scene is often desired. Although custom methods have been proposed for specialized scenarios [5, 21, 24, 29, 33, 34], a standard method for different purposes and under arbitrary conditions is yet to be presented.

To overcome this problem, both sensors and algorithms must be designed for long-term persistence under varying, real-world conditions. On the sensor side, one solution is to supplement the traditional visible light camera with other sensor types. Such multi-sensor systems are more persistent to changes in the environment; if the output of one sensor is impaired due to sub-optimal conditions, other sensor types are not necessarily affected.

Consequently, a special interest in thermal infrared cameras has recently developed. Thermal cameras cannot capture visible light, but only pick up the

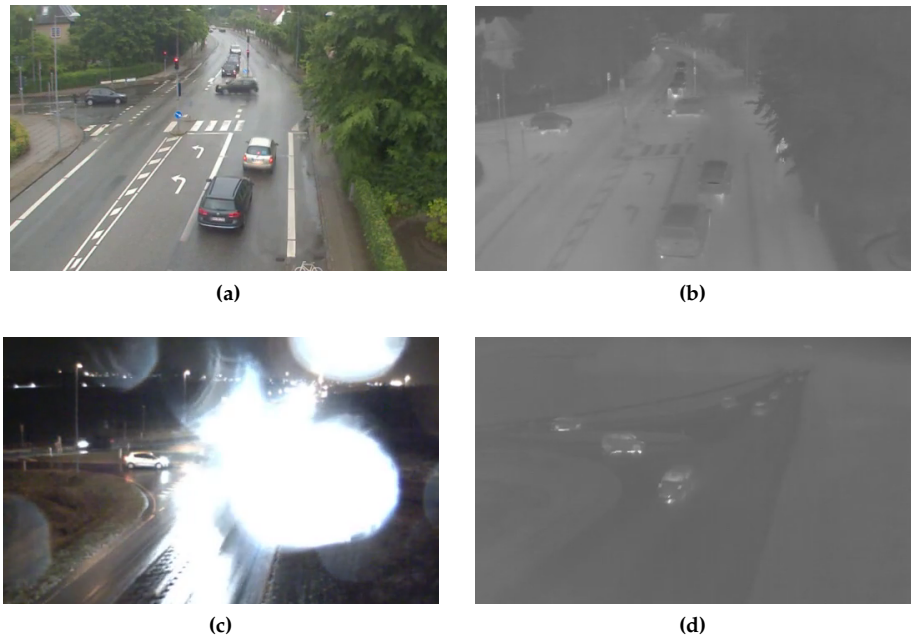


Fig. E.1: RGB and thermal images of two scenes. In the top scene, the RGB image (**a**) provides a detailed view of the road users. The thermal image (**b**) shows fewer details, but gives a better view of the pedestrian behind the tree on the pavement. In the bottom scene, the headlights of the approaching vehicles blurs parts of the RGB image (**c**) and introduces glare by the raindrops on the lens. Fortunately, the corresponding thermal image (**d**) is unaffected by the headlights.

infrared radiation emitted by objects [8]. The infrared radiation depends on the temperature of the object, thus making the imaging system independent of illumination and less dependent on visual obstructions caused by, e.g., fog or rain. As seen in Figure E.1, the downside is that thermal images are less detailed and provide an unfamiliar visual impression to a human observer. When combined, traditional visible light (RGB) cameras and thermal cameras enables 24-h surveillance under arbitrary lighting conditions and improve the observability under challenging environmental conditions.

In order to utilize the information from the various sensors, one should fuse the information at some point in the data processing chain. However, how is the data fusion actually performed? When fusing the data streams, how should the different streams be weighted against each other? In the ideal case, the weights are dependent on the information quality of the data stream of a particular sensor, e.g., how objects of interest are distinguished from other parts of a scene. The information quality of a data stream is dependent on the sensor attributes, the object nature, scene geometry and environmental conditions, but also on the purpose and nature of the subsequent analysis of

the data.

In this paper, we present a novel method for the context-based fusion of video from thermal and RGB video streams. The context-based fusion is integrated with the segmentation of the scene, which is the first and crucial step of bottom-up processing pipelines [3] commonly used in real-time surveillance systems. We integrate the contextual information of the scene to assess the quality of the video data, which we use to fuse the output of two parallel segmentation pipelines.

The methodology of image fusion and related work is discussed in Section 2. In Section 3, we deduce context-based quality parameters based on environmental conditions and the appearance of the video data. These parameters are used to design a context-adaptive fusion pipeline, which is described in Section 4. This pipeline is exemplified using an image segmentation algorithm in Section 5 to create a fused, segmented image, which is common to both the thermal and RGB video streams. *Subsequently, we present extensions for the application of traffic monitoring in Section 6* [2]. In Section 7, we evaluate the context-based fusion on our own and two commonly-used datasets against competing approaches to image fusion. Finally, our conclusions are presented in Section 8.

2 Related Work

Different sensors have advantages and disadvantages in terms of further processing. *To overcome the individual downsides of different sensors, multimodal systems have been developed. These systems use information from multiple sensors and information sources to combine and enrich the available data* [2]. The potential of these methods, especially for traffic surveillance, has been emphasized by Buch et al. [3]. In this section, different fusion approaches will be presented and discussed. *The main focus will thus be on the fusion of video data from thermal and RGB cameras* [2].

Fusion approaches are generally divided into three levels: pixel-level fusion, feature-level fusion and decision-level fusion, depending on the stage at which the fusion takes place [10] [2].

Decision-level fusion combines the output from two or more parallel processing pipelines. The results are merged by Boolean operators or the weighted average. Serrano et al. [25] perform parallel segmenting of thermal and RGB data and select the representative output on the basis of confidence heuristics [2].

Feature-level fusion takes place one step earlier in the processing pipeline. Features from all input images are extracted individually and then fused into a joint feature space. Kwon et al. [17] used this technique for automatic target recognition [2].

Pixel-level fusion is the most common approach. In this type of fusion, the input images are merged into one. Details that might not be present in one image are hereby

added by the other modality. Common examples are structures occluded through dark shadows or smoke in RGB images that are revealed with the help of a thermal image. Pixel-level fusion requires all input images to be spatially and temporally aligned. This alignment, also called registration, is a challenge. Automatic image registration approaches often fail as there is no correlation between the intensity values of the modalities [6]. A common approach is to manually select corresponding points in both modalities and compute a homography. However, special-case automatic methods exist; these use features that are most likely present in both modalities, e.g., contours [12], Harris corners [13] or Hough lines [14] [2].

Shah et al. [26] perform the fusion after different wavelet transforms of the images. This allows a fusion rule based on frequencies rather than pixels [2]. The approach preserves the details while still reducing artifacts. Chen and Leung followed a statistical approach in [4] by using an expectation-maximization algorithm.

Lallier and Farooq [18] perform the fusion through adaptive weight averaging. The weight per pixel is hereby defined by the number of equations that express the interest in the specific pixel. In the context of this work, these are the degree to which an object is warmer or colder in the thermal domain, the occurrence of contrast differences and *large spatial and temporal intensity variations in the visual domain* [2].

Instead of fusing the images to a new image, which can be represented in RGB, other methods simply combine the inputs into a new format. St-Laurent et al. [27] adapt a Gaussian Mixture Model (GMM) algorithm for extracting moving objects to work with "Red-Green-Blue-Thermal" (RGBT) videos [2]. In this way, important information is automatically revealed by the object extraction algorithm.

3 Context-Based Image Quality Parameters

In this work, we use a pixel-level fusion approach. However, unlike usual pixel-level approaches, the RGB and thermal images are not fused immediately. Instead, we use the soft segmentation results from individual processing of the thermal and RGB video streams. The quality of the video streams is used to fuse the soft segmentation results and, thus, forms a context-aware, quality-based fusion.

In the following, we discuss the conditions that effect the image quality for surveillance scenarios and how those conditions may be predicted by data from different sources. The aim is to construct context-sensitive indicators, q_{RGB} and q_{thermal}, that express the usefulness of each modality.

When assessing the relative qualities of the thermal and RGB images, we distinguish between predictable and unpredictable conditions. The predictable conditions are considered "static" under short time spans, but may change gradually over several hours, such as the position of the sun or the general weather conditions. The unpredictable conditions cannot be measured

(a) **(b)** **(c)** **(d)**

Fig. E.2: Thermal images of the same scene with different entropy values. (**a**) $H = 4.23$; (**b**) $H = 5.04$; (**c**) $H = 6.65$; (**d**) $H = 7.67$. Adapted from [2].

beforehand and may change rapidly in a few seconds, for example when a cloud temporarily blocks the Sun. In the following, we start by discussing the predictable conditions in terms of the thermal and RGB images, which is followed by a discussion of the unpredictable conditions.

3.1 Predictable Thermal Image Quality Characteristics

Thermal cameras measure the infrared radiation emitted by all objects. The energy of the radiation mainly depends on the object temperature [2]. A constant factor, referred to as emissivity, scales the radiation for different materials [8]. If the emissivity is known, the temperature of objects obtained in thermal images can be calculated using the Stefan–Boltzmann law [30]. *However, Automatic Gain Control (AGC) often forms part of many of the thermal cameras that are built for surveillance, and this implies that the exact relation between radiation energy and intensity values is often unknown* [2].

Objects consisting of different materials have different intensity values in a thermal image, even if they have almost the same temperature [8]. Typical scenes consist of several different materials, and we therefore expect a certain amount of information in the thermal image; also for scenes without foreground objects. If no objects can be distinguished, the information content is low. Consequently, the image entropy can be used as a quality indicator for thermal images. The entropy, H, is defined as [2]:

$$H = -\sum_{i=0}^{255} p(I_i) \frac{\log(p(I_i))}{\log(2)} \tag{E.1}$$

where $p(I_i)$ is the percentage of pixels with intensity i in the thermal image I.

Figure E.2 shows a side-by-side comparison of the same location at different times. The right images appear much more detailed and, therefore, of higher quality. The corresponding entropy values correlate with this impression [2].

Experiments have been conducted and have shown that a linear function enforces a too strong down-rating of low entropy values. *Thus, a sigmoid function is found to be a better approximation of the mapping function between the*

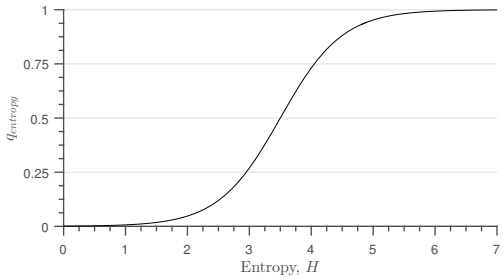


Fig. E.3: Shape of the q_{entropy} quality function relative to the entropy of the thermal image. From [2].

entropy and quality of the thermal image [2]. Thus, the entropy quality parameter is defined as:

$$q_{\text{entropy}} = \frac{1}{1 + e^{(3.5-H)\cdot 2}} \tag{E.2}$$

The shape of q_{entropy} is shown in Figure E.3.

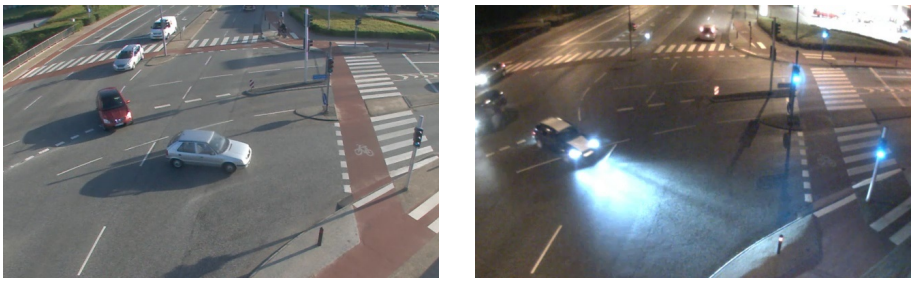
3.2 Predictable RGB Image Quality Characteristics

The predictable image quality of an RGB image is closely correlated with the amount of light in the scene. When working with outdoor scenes, the amount of light in the scene is strongly dependent on the available sunlight. The more sunlight, the higher the image quality. However, in full sunlight, shadows will appear, which might be the cause of false positives when segmenting the image. The state of the weather in a scene is pivotal when estimating the general observability of the scene. Phenomena such as mist and fog reduce the visibility. Rain and snowfall introduce spatio-temporal streaks in the image, which further impedes the view.

In the following, we will discuss the effect of these phenomena on the RGB image quality.

Illumination

Figure E.4 shows the same scene in the afternoon and at dusk. *While a human being can easily label the cars in the scene, segmentation algorithms would be highly disturbed by the large shadows and reflections. Although several shadow suppression algorithms exist nowadays [22], shadows still disturb the detection process* [2]. The handling of reflections, as imposed by moisture and shiny surfaces, is still an unsolved problem. In conclusion, both images presented in Figure E.4 should be rated as low quality, although the reasons for the low quality are different and so may the quality rating be.



(a) **(b)**

Fig. E.4: RGB images with common challenging conditions. (**a**) Shadows; (**b**) Reflections and halos. From [2].

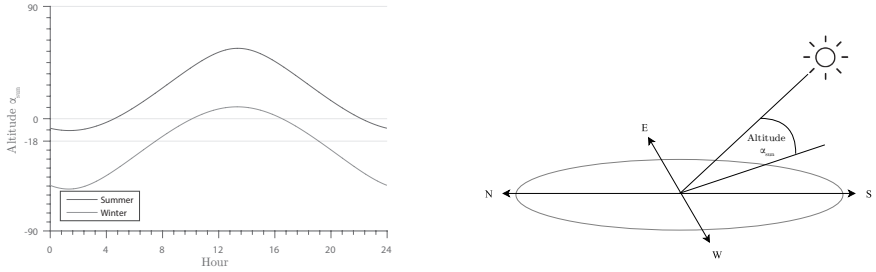
Consequently, images with low light conditions, such as twilight and night, should be rated as low quality [2]. The elevation angle of the Sun, as illustrated in Figure E.5, can be used as an input parameter. The solar elevation angle, α_{sun}, is defined as the angle between the ground plane and the Sun's position vector; see Figure E.5b. It is dependent on the longitude and latitude of the scene, as well as the date and time of the recording [19]. The Sun is visible for angles $\geq 0°$. In practice, however, noticeable illumination is not present before $-6°$, known as civil twilight [23]. Additionally, as soon as the Sun is visible, the illumination condition is not perfect. Therefore, in this work, we require the altitude of the Sun to increment an additional $6°$ before we define good illumination and, thus, set $q_{\text{sun}} = 1$. If the altitude of the Sun is below $-6°$, the Sun does not contribute to the light in the scene, and we set $q_{\text{sun}} = 0$. However, there might be other light sources that contribute to the illumination of the scene, for example street lights. Thus, we define a non-zero minimum quality parameter, q_{smin}. This leads to the following formula:

$$q_{\text{sun}} = \begin{cases} 1.0 & \text{if } \alpha_{\text{sun}} \geq 6° \\ \max\left(\frac{\alpha_{\text{sun}}+6°}{12°}, q_{\text{smin}}\right) & \text{if } -6° \leq \alpha_{\text{sun}} < 6° \\ q_{\text{smin}} & \text{if } \alpha_{\text{sun}} < -6° \end{cases} \tag{E.3}$$

with the solar elevation angle α_{sun} and a minimum quality parameter q_{smin}, which is set according to the amount of artificial light available in the scene. The resulting function is displayed in Figure E.6 with $q_{\text{smin}} = 0.2$.

Shadows

Two external factors specify the occurrence of the shadows. First of all, shadows may appear only on sunny days. Sunny days may be detected by integrating a weather station next to the setup or by accessing a weather



(**a**) (**b**)

Fig. E.5: Solar altitude over a day in the summer and winter (**a**); the solar altitude is defined by the angle between the ground plane and the Sun's position vector (**b**). From [2].

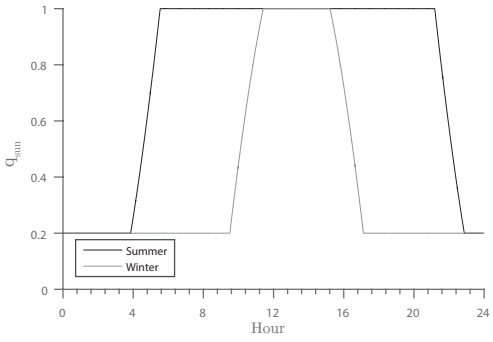


Fig. E.6: Development of the q_{sun} quality indicator over a winter and summer day. The corresponding altitude of the Sun is shown in Figure E.5a.

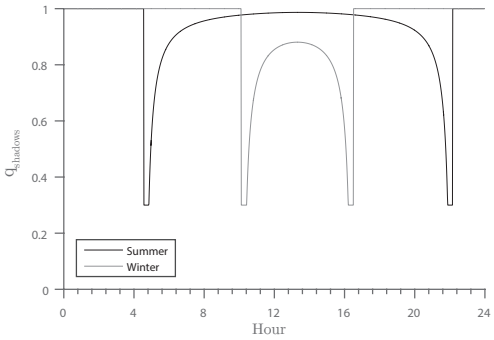


Fig. E.7: Development of the q_{shadows} quality indicator over a day during summer and winter. The corresponding altitude of the Sun is shown in Figure E.5a.

database. *The length of these shadows is determined by the Sun's position. Therefore, both weather data and the solar elevation angle must be considered to present a model showing to what extent cast shadows might be present in the scene. The length of shadows can be calculated through:* [2]

$$L = h/tan(\alpha) \tag{E.4}$$

with h being the object height [2]. With unit object height, Equation (E.5) can serve as a quality function, where ψ is a scaling factor, q_{weather} is the weather quality indicator defined in Section 3.2 and q_{shmin} is the minimum required quality.

$$q_{\text{shadows}} = \begin{cases} \max(1 - \psi L, q_{\text{shmin}}) & \text{if } \alpha_{\text{sun}} > 0 \wedge q_{\text{weather}} = 1 \\ 1.0 & \text{otherwise} \end{cases} \tag{E.5}$$

The resulting function is plotted in Figure E.7 with $q_{\text{shmin}} = 0.3$ and $\psi = 50$.

Weather Conditions

Different weather conditions may harm segmentation algorithms through various phenomena, such as mist, fog, rain and snow. The long-term effects of rain are visible as reflections in puddles and moisture on the road. A quantitative rating, however, is not so easily derived. For this work, we have grouped weather conditions obtained from [20] into five broad categories, as seen in Table E.1. A clear sky is defined as optimal conditions with a quality rating of one. Clouds and light mist reduce the amount of light available in the scene and are as such assigned a lower quality rating of 0.8. The occurrence of rain and snow induces spatio-temporal noise and reduces the visibility of the

Weather Condition [20]	Category	q_{weather}
Clear	Good conditions	1.0
Overcast Cloudy Light mist, drizzle	Low/varying illumination	0.8
Heavy drizzle, mist Light rain	Reflections/moisture	0.6
Snow Hail Heavy rain Thunderstorm	Particle occlusion/precipitation	0.3
Fog, haze Dust, sand, smoke	Reduced visibility	0.3

Table E.1: Weather conditions and their corresponding category and quality indicator, q_{weather}. Adapted from [2].

scene. We distinguish between light and heavy rain and all other types of precipitation. The spatio-temporal effects of light rain are barely visible, whereas raindrops may be visible during heavy rain, snow and hail [9]. Fog and haze do not occur as spatio-temporal effects, but greatly reduce the visibility and are thus grouped with heavy rain and snow.

3.3 Unpredictable Image Quality Characteristics

We define unpredictable conditions as rapidly changing, dynamic conditions that may not be predicted by the sensors or the available contextual knowledge. In the RGB image, this includes rapidly changing illumination, for instance caused by clouds that temporarily blocks the Sun. In the thermal image, the most prominent, dynamic change is caused by the auto-gain mechanism of the thermal camera. The auto-gain automatically maximizes the contrast of the thermal image by adjusting the gain of the camera, which means that the appearance of a scene may change suddenly when cold or warm objects enter the scene.

Because the rapidly-changing conditions may not be predicted beforehand, we will rate them by their effect on the subsequent image segmentation process. Typically, most segmentation algorithms will respond to rapidly-changing conditions with abrupt changes in the ratio of Foreground (FG) and Background (BG) pixels. Over time, the segmentation algorithm will incorporate the changes, and the ratio of FG and BG pixels stabilizes.

We can incorporate this characteristic in a quality indicator, such that rapid changes in the FG/BG ratio are penalized. This indicator, q_{fg}, is

defined in Equation (E.6), where τ defines the average foreground ratio and γ is a weight controlling the foreground deviation:

$$q_{\text{fg}} = \max(1 - \gamma(r_{\text{fg}} - \tau), 0) \tag{E.6}$$

where the current foreground ratio, r_{fg}, is defined as:

$$r_{\text{fg}} = \frac{1}{XY} \sum_{x=1}^{X} \sum_{y=1}^{Y} \mathbb{1} \tag{E.7}$$

where $\mathbb{1}$ denotes an indicator function that returns one if the image at position (x,y) is foreground, otherwise zero, and (X, Y) are the image dimensions.

The q_{fg} indicator is computed separately for the RGB and thermal image streams.

3.4 Combined Quality Characteristics

At this stage, we have developed several indicators for the image quality of both modalities, which should be combined into one quality indicator for each modality. We start by combining the indicators that correspond to the static predictable conditions. In the thermal domain, this is easy, as there is only one indicator, q_{entropy}, to consider. In the RGB domain, the quality indicators are closely interrelated. However, the exact nature of these relations is unknown, and a study of this is beyond the scope of this work. Therefore, we here assume decorrelation and hence combine the indicators by multiplication:

$$q_{\text{static}_{\text{RGB}}} = q_{\text{sun}} \cdot q_{\text{shadows}} \cdot q_{\text{weather}} \tag{E.8}$$

$$q_{\text{static}_{\text{Thermal}}} = q_{\text{entropy}} \tag{E.9}$$

The predictable and unpredictable quality indicators are combined for each modality by taking the minimum value:

$$q_{\text{RGB}} = \min(q_{\text{fg}_{\text{RGB}}}, q_{\text{static}_{\text{RGB}}}) \tag{E.10}$$

$$q_{\text{Thermal}} = \min(q_{\text{fg}_{\text{Thermal}}}, q_{\text{static}_{\text{Thermal}}}) \tag{E.11}$$

To prevent artifacts, the quality indicators are gradually updated:

$$q_t = \begin{cases} q_t & \text{if } q_t \leq q_{t-1} \\ \alpha q_t + (1 - \alpha) q_{t-1} & \text{otherwise} \end{cases} \tag{E.12}$$

where α is the update rate of the segmentation model. The calculation is performed independently in the RGB and thermal domain.

4 Context-Based Fusion

The following section presents a new approach to fusing the image streams by integrating the quality indicators into a segmentation pipeline. As opposed to other works, we do not fuse the input data directly. Rather, we have used the intermediary results of two parallel segmentation algorithms. The results are weighted in accordance with the quality indicators described before to ensure that the system is context-aware. Figure E.8 illustrates the basic principle of this work. The core contribution is illustrated in Part II, object identification, of Figure E.8. The images of the thermal camera are registered into the coordinate system of the RGB image by using a planar homography [11] such that positions on the road plane in the thermal image correspond to the same positions in the RGB image. The registered images are fed into two parallel segmentation algorithms from which we get the intermediate, soft segmentation results that represent, for each pixel, the degree of belief that the pixel is considered to be in the foreground. In this work, we denote this as the distance maps. The fusion of these maps is discussed in the following. The details of Part III, distance modulation, of Figure E.8 are explained in Section 6.

We normalize the quality indicators q_{RGB} and q_{Thermal} to add up to one and use the normalized values as weights for the adaptive fusion of the distance maps. The weights are calculated as follows:

$$w_{\text{RGB}} = \frac{q_{\text{RGB}}}{q_{\text{RGB}} + q_{\text{Thermal}}} \qquad w_{\text{Thermal}} = \frac{q_{\text{Thermal}}}{q_{\text{RGB}} + q_{\text{Thermal}}} \tag{E.13}$$

The distance map of each modality is multiplied by its corresponding weight, and the results are summed to create a unified, fused distance map:

$$D_{\text{F}} = w_{\text{RGB}} D_{\text{RGB}} + w_{\text{Thermal}} D_{\text{Thermal}} \tag{E.14}$$

At this stage, small inaccuracies in the spatial and temporal registration can be compensated. A simple mean filter applied on the fused distance map dissolves the pixel grid and therefore fuses information from neighboring pixels [2].

The final step in the segmentation is the decision as to whether a pixel is defined as foreground or background. We threshold the fused distance map on a per-pixel level:

$$\text{FG} = \begin{cases} 1 & \text{if } D_{\text{F}} \geq T \\ 0 & \text{otherwise} \end{cases} \tag{E.15}$$

where T is the segmentation threshold, usually set to one.

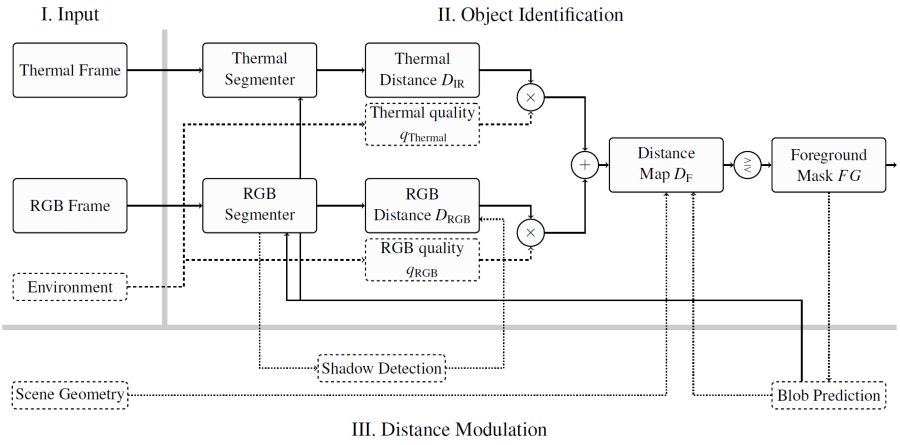


Fig. E.8: System design overview: The three main phases of the fusion algorithm are illustrated. Two registered input streams are processed by two parallel segmentation algorithms. The soft segmentation results from these algorithms, denoted as distance maps, are fused by using the quality indicators of each stream. Distance modulation functions may improve the algorithm for the purpose of traffic monitoring by using constraints derived from scene geometry, shadow detection and object (blob) detection; see Section 6. Adapted from [2].

5 Segmentation Algorithm

In the framework presented in the previous section, we fused the intermediate output of two segmentation algorithms. Any image segmentation algorithm may be used, as long as it generates a soft-decision pixel map that may be used as the distance map of Figure E.8. In the rest of this paper, we apply a particular segmentation method in order to be able to quantify the benefits of the proposed fusion strategy. We use the classic Gaussian Mixture Model (GMM) [28] to exemplify our context-fusion framework.

The GMM is widely used within the domain of traffic surveillance [3] and represents a well-known platform to showcase the context-based fusion. A brief introduction to the GMM is given in the following.

During the calculation of the background distance based on the GMM, each pixel is tested against each component of the GMM's background model. The Mahalanobis distance of the sample value from the background model is hereby the determining factor for acceptance [2]. A pixel x at time t is defined to match the background component if it falls within λ standard deviations:

$$M_{i,t} = \left(\frac{|x_t - \mu_{i,t-1}|}{\lambda \sigma_{i,t-1}} < 1 \right) \tag{E.16}$$

where $M_{i,t}$ is the i-th background model at time $t+1$, $\mu_{i,t-1}$ and $\sigma_{i,t-1}$ is the mean value and standard deviation of $M_{i,t-1}$, respectively.

The mean and standard deviation of the background models are constantly updated as follows:

$$\mu_{i,t} = (1-\beta)\mu_{i,t-1} + \beta x_t \tag{E.17}$$

$$\sigma^2_{i,t} = (1-\beta)\sigma^2_{i,t-1} + \beta(x_t - \mu_{i,t-1})^2 \tag{E.18}$$

where β is defined as:

$$\beta = \alpha\mathcal{N}(x_t, \mu_{i,t-1}, \sigma^2_{i,t-1}) \tag{E.19}$$

and α is a constant update rate.

The acceptance distance of the sample as the foreground in Equation (E.16) is normalized by the specific variance $\sigma_{i,t}$ and the threshold value λ. Large distance values indicate a high probability of the pixel being in the foreground, whilst small values show high conformity with the component. With this in mind, an approximation of the general conformity of a pixel in the model can be expressed by computing the distance value, D_t: [2]

$$D_t \approx \begin{cases} d_{0,t} & \text{if } M_{0,t} \\ d_{1,t} & \text{if } M_{1,t} \\ & \dots \\ d_{b,t} & \text{if } M_{b,t} \\ \min(d_{0,t}, d_{1,t}, \dots, d_{b,t}) & \text{otherwise} \end{cases} \tag{E.20}$$

with:

$$d_{i,t} = \frac{|x_t - \mu_{i,t-1}|}{\lambda\sigma_{i,t-1}} \tag{E.21}$$

where b denotes the total number of background models.

If a match $M_{i,t}$ is found, the corresponding value of $d_{i,t}$ is used to express the distance. Otherwise, the distance to the closest component is used. The resulting values of all pixels form a map expressing the deviation of image regions from the background, and this is fed into the context-based fusion framework as the distance map [2]. Figure E.9 displays the distance maps, their fusion and the effect on the resulting mask.

6 Application to Traffic Monitoring

The preceding sections described the main contribution of this work. In the following, we will present specific extensions for traffic surveillance to show the modularity of the proposed algorithm. In Figure E.8, these extensions are categorized as III, distance modulation.

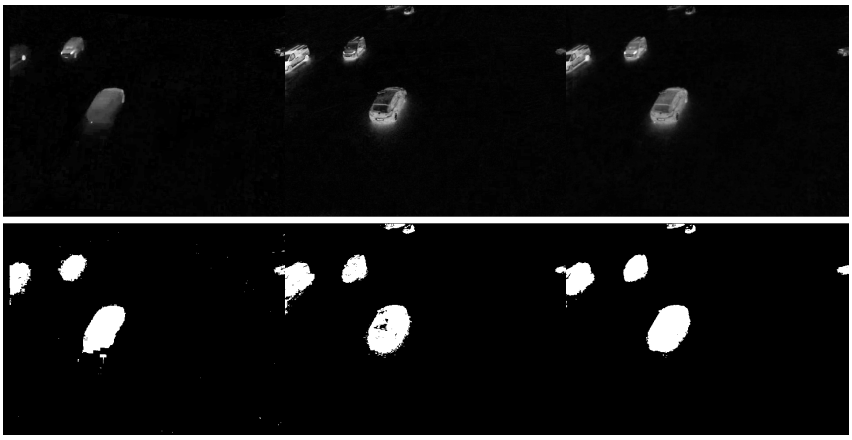


Fig. E.9: Distance maps of the different modalities and results after thresholding. The intensity of a pixel corresponds to the distance to the background model of each parallel segmentation algorithm. Bright pixels indicate a high probability for foreground objects. (**a**) Thermal; (**b**) RGB; (**c**) fused. Adapted from [2].

6.1 Shadow Detection

A common extension of background modeling techniques is shadow detection. Shadows of intruding objects do not match the background model; they appear as darker formerly-illuminated areas and are, therefore, defined as foreground. Depending on the purpose of the system, the labeling of shadow areas as foreground is a false positive error. In most surveillance scenarios, only the objects (not their shadow) are of interest [22] [2].

Prati et al. [22] distinguish between deterministic approaches to shadow detection, which use an "on/off decision process", and statistical approaches, which "use probabilistic functions to describe the class membership". However, both methods can fail and lead to false negatives, just as false positives may also occur. *If the main task is to identify all foreground objects, as in the case of traffic surveillance, especially false positives may harm the results. Whole objects may be classified as shadows. To address this issue, shadow areas have been pruned rather than completely removed in this work* [2].

State-of-the-art methods perform a labeling function in the resulting foreground mask. Instead of making this hard decision, the distance related to the areas marked as shadows is scaled down. In this work, a fixed scaling method has been used. A scaling based on the shadow certainty may be a possible extension. As the background distance correlates with the certainty of a pixel being in the foreground, the downscaling may be considered as bringing uncertainty to the decision [2]. Consequently, the decision of whether a pixel is defined as shadow is only made indirectly when deciding

whether the pixel is categorized as either foreground or background.

The subsequent fusion of the modalities is the important step for this method to work. Objects that have also been found in the thermal image are most likely found anyway, and shadows are voted further down as they are not present in the thermal domain. Especially small areas of false positives can be recovered as being a foreground object using this technique. The mean filter subsequent after the fusion helps the process of removing outliers. Additionally, the quality indicators allow prediction of scenes with shadows. Therefore, the process can be triggered to be context aware [2].

6.2 Blob Prediction

A successful segmentation algorithm for traffic surveillance must handle the different speeds of the traffic, which implies that all objects *must be handled as foreground even when staying in the scene for a longer time. For this purpose, the blob prediction method proposed by Yao and Ling [31] has been integrated in this work* [2]. The position of foreground blobs is predicted for each frame, and the update rate α of the segmentation algorithm is significantly lowered for these areas. *Consequently, objects must stay for a very long time before merging into the background* [2].

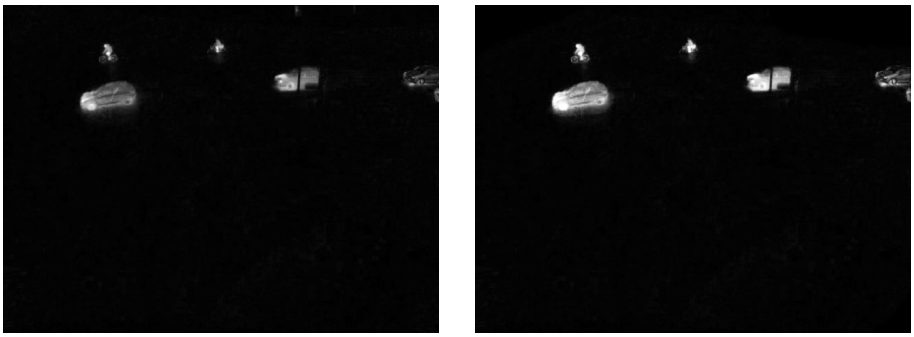
To predict blob positions for the current frame, t, blobs from t and $t-1$ are matched. Subsequently, the displacements between t and $t-1$ are applied on t. The matching is done with a nearest neighbor search of the blob's centroids. If no neighbor within range ρ is found, the blob is supposed to be stationary as no prediction about the movement can be made [2].

We extend the method by Yao and Ling [31] by dilating the predicted blobs and smoothing out edges. *This is done to prevent artifacts in the background model caused by inaccuracies in the blob prediction. The update rate α of the segmentation algorithm is thus calculated as:* [2]

$$\alpha = D_{\text{predict}}\alpha_{\mathbf{fg}} + (1 - D_{\text{predict}})\alpha_{\mathbf{bg}} \tag{E.22}$$

where $0 \leq D_{predict} \leq 1$ indicates the value in the blob prediction image and $\alpha_{\boldsymbol{fg}}$ and $\alpha_{\boldsymbol{bg}}$ are the update rates for foreground and background regions, respectively [2].

Another purpose of the blob prediction is presented in this work. As the boundary of foreground objects changes only gradually, the predicted blobs provide a very good estimate of the foreground of the next frame. This can help the segmentation, as it is more likely to locate an object where predicted than elsewhere in the scene. Objects follow a trajectory and generally do not appear unexpectedly [2]. To express this characteristic, another modification of the distance map is performed. Analogous to the shadow suppression, the predicted areas are up-scaled in the distance map. Figure E.10 demonstrates the effect. The right image of Figure E.10 shows the distance map after the blob prediction. Compared to the distance map before the prediction, as shown on the left of Figure E.10,



(a) (b)

Fig. E.10: Distance map before (**a**) and after blob prediction-based modulation (**b**). From [2].

one sees that objects appear brighter in the right image and thus have a higher likelihood of being declared foreground.

6.3 Scene Geometry-Based Knowledge

The principle presented in the last sections can be used for another constraint. By looking at the scene geometry, one can easily divide the image into three classes. The first class of pixels is areas where no foreground is expected under any circumstances, for example trees or the sky. The second class of pixels denotes the areas into which objects may move. A sudden appearance of objects is unlikely or even excluded, but objects may move to these areas from other parts of the image. These areas are referred to as neutral zones. The last class describes the areas in which we expect foreground objects to appear. These areas are called entrance areas in the following. Entrance areas can normally be found at the borders of the image as objects enter the scene, normally from outside the viewport of the camera. Objects may, however, also reappear after occlusion or enter from occluded areas. Based on this classification, a mask can be drawn as seen in Figure E.11 [2].

Firstly, excluded areas cannot be categorized as foreground when the corresponding values in the distance map are set at zero. *Secondly, the distance values for neutral zones are scaled down by* $s_{neutral}$ *to make it less likely to find foreground pixels in these areas. This is possible because the blob positions have been predicted and uprated beforehand* [2]. Areas to which we expect objects to move are untouched afterwards or even uprated, whereas unpredicted regions are down-rated. This helps remove noise, and found objects are considered more reliable.

Fig. E.11: Scene area classes. Green: entrance areas; red: excluded areas; rest: neutral. From [2].

7 Experiments

A series of experiments has been conducted to evaluate both the quantitative and the qualitative performance of the proposed algorithm. *This section begins with an elaboration about the datasets that have been used in this work, followed by a description of the performance metrics and the results of the experiments. Finally, an in-depth analysis of the qualitative performance is presented* [2].

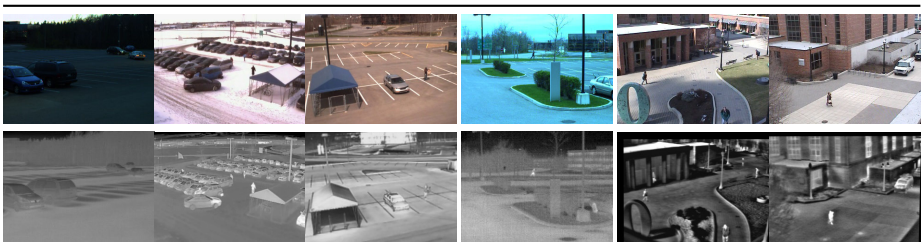
7.1 The Datasets

The main dataset used in this work contains a large number of thermal-RGB recordings of intersections in Northern Jutland, Denmark, recorded during 2013. The videos are undistorted using the line-based parameter estimation by Alemán-Flores et al. [1] [2]. To be able to benchmark the proposed algorithm, we include two commonly-used datasets. *The Ohio State University (OSU) Color-Thermal Database [7] of the Object Tracking and Classification Beyond the Visible Spectrum (OTCBVS) Benchmark Dataset Collection contains RGB and thermal data of two surveillance scenarios. The videos contain pedestrians recorded on the campus of Ohio State University. The National Optics Institute (INO) Video Analytics Dataset (`http://www.ino.ca/en/video-analytics-dataset/`) contains a set of multimodal recordings of parking lot situations, including data on cars, cyclists and pedestrians* [2].

As we know the exact location and time of our own datasets, we can compute the altitude of the Sun directly and retrieve weather information from a nearby weather station. As this contextual information is not known for the external datasets, we derive the contextual information from manual scene observations. Weather conditions are grouped into the categories introduced in Table E.1. All scenes tested during the experiments are listed in Tables E.2 and E.3. The contextual information for each scene is listed in Table E.4.

Day	Night	Auto Gain	Heavy Rain	Snowing

Table E.2: Test scenes from our own dataset. The videos are rectified using a line-based parameter estimation method [1]. From [2].



INO ParkingEvening	INO ParkingSnow	INO CoatDeposit	INO TreesAndRunner	OTCBVS 3	OTCBVS 4

Table E.3: Test scenes from the benchmark datasets. From [2].

Sequence	Annotated Frames	Average Number of Objects per Frame	Weather Classification	q_{weather}	Sun Altitude	q_{sun}	q_{shadows}
Day	70	6.4	Good conditions	1.0	20°	1.0	0.95
Night	70	6.9	Low illumination	0.8	−19°	0.20	1.0
Auto Gain	180	9.0	Moisture	0.6	20°	1.0	1.0
Heavy Rain	70	6.7	Moisture	0.6	29°	1.0	1.0
Snowing	70	5.6	Precipitation	0.3	9°	1.0	1.0
INO ParkingEvening	70	2.1	Good conditions	1.0	−12°	0.20	1.0
INO ParkingSnow	70	7.0	Low illumination	0.8	86°	1.0	1.0
INO CoatDeposit	70	2.8	Low illumination	0.8	46°	1.0	0.98
INO TreesAndRunner	70	1.0	Low illumination	0.8	0°	0.50	0.30
OTCBVS 3	70	3.9	Low illumination	0.8	12°	1.0	0.91
OTCBVS 4	70	1.0	Good conditions	1.0	12°	1.0	1.0

Table E.4: Annotation properties and corresponding context-based quality characteristics for each test scene. For our own dataset, Sun altitude and weather information are provided through direct computations and a weather database, respectively. For the benchmark dataset, this information has been derived from manual scene observations.

7.2 Performance Metrics

We evaluate the experiments by the quantitative performance metrics used in [16]. These metrics are the Detection Rate (DR) and the False Alarm Rate (FAR), defined as:

$$\mathrm{DR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{E.23}$$

$$\mathrm{FAR} = \frac{\mathrm{FP}}{\mathrm{TP} + \mathrm{FP}} \tag{E.24}$$

with True Positives (TP), False Positives (FP) and False Negatives (FN). The DR is also known as recall or the true positive rate and describes the sensitivity of a detector. The FAR corresponds to $1 - p$*, where* p *is the detector's precision or specificity* [2].

In order to evaluate the performance metrics, access to the true data, commonly referred as the Ground Truth (GT), is needed. GT must be created manually and is a laborious task. Thus, only a small sample of the results can be tested. In this work, 70 successive frames have been annotated for each test set with the exception of 180 annotated frames of the Auto Gain set [2]. In our own dataset, this amounts to approximately 3 and 7 s of video, respectively. The GT has been annotated using the Aalborg University Visual Analysis of People (AAU VAP) Pixel Annotator (`https://bitbucket.org/aauvap/multimodal-pixel-annotator`) where the boundary of each object has been traced manually by a mouse. The average number of objects per frame is shown for each sequence in Table E.4.

7.3 Quantitative Results

In order to evaluate the performance of the proposed method, extensive experiments have been performed and evaluated with the described performance metrics. Besides the algorithm itself, each dataset has been processed by applying four alternative strategies, presented below: [2]

- RGB: individual processing of the RGB modality by the proposed method.
- Thermal: individual processing of the thermal modality by the proposed method.
- RGBT: pixel-wise, naive (not context-aware) fusion of RGB and thermal streams.
- Select: confidence-based selection as presented by Serrano-Cuerda et al. [25]

All strategies are based on the GMM background segmentation algorithm presented by Stauffer and Grimson [28] and improved by Zivkovic [32] and

differ only in the ways the data fusion is performed. This allows us to measure the contribution of our context-aware fusion approach compared to other fusion approaches or single-modality processing. As mentioned in Section 5, other segmentation algorithms may be used in combination with the proposed method. By using a well-known approach, such as the GMM, however, we believe that the comparison reveals interesting insights on the strengths and weaknesses of the proposed method.

This work aspires to create a system that works without requiring the manual tuning of its parameters for different conditions. *Therefore, only the learning time for each scene has been adjusted to match the specific situation. For example, scenes with much traffic need more time to learn a stable background model. For the case of the presented algorithm, background models have been learned individually before the described adjustments were made* [2]. This procedure is necessary because the predicted foreground regions are learned slowly, and false positives are very likely to appear during the learning phase.

The update rate α of the segmentation algorithm has been set to be slower for the alternative strategies. As the GMM background modeling does not differ between foreground and background in the update step, a quick update rate would result in foreground objects merging into the background. This is also the case in the learning phase of the proposed method. Consequently, the same α has been used here. All important experimental parameters are listed in Table E.5, where the parameters below the line apply only for the proposed method [2].

Shadow detection has been performed for all experiments containing RGB data [2]. Pixels that have been categorized as shadow have been classified as background in the reference methods. Furthermore, the scene area classes have been applied on the resulting data. *This approach ensures that equal conditions have been created for all strategies, and identified differences in the results of the proposed algorithm in contrast to the alternative strategies can be explained by its core contributions* [2].

The results of the experiments are displayed in Table E.6. The general performance of the proposed algorithm can be considered very good due to a average DR and FAR of 0.95 and 0.35, respectively. The table clearly shows that the goal of creating a robust method for a wide bandwidth of conditions has been achieved. Only the proposed method shows good performance for every test sequence, which is expressed in the average FAR and DR rates, which are significantly better than the alternative strategies. These strategies fail in different scenarios, but show better performance than the proposed method for some scenarios. The reasons for this are manifold and will be discussed in Section 7.4 [2].

As expected, all fusion approaches generally tend to demonstrate better performance than single-modality methods. The method presented by Serrano-Cuerda et al. [25] also seems to perform well at first glance. However, when analyzing the results in detail, it becomes clear that the results are, at best, as good as one of the single modalities. This is related to the design of the algorithm; it is designed to select one

Parameter	Value	Description
α	0.0005	GMM update rate
K	5	Number of components for GMM
λ	4	Number of standard deviations for background acceptance for GMM
T	1	Segmentation threshold of the distance map
α_{BG}	0.0033	Background update rate for blob-based prediction
α_{FG}	0.000033	Foreground update rate for blob-based prediction
τ	0.1	Foreground ratio
γ	5.0	Foreground deviation weight
ρ	17	Blob match radius (px)
s_{shadow}	0.3	Distance scaling factor for shadow regions
s_{predict}	1.5	Distance scaling factor for predicted regions
s_{neutral}	0.5	Distance scaling factor for neutral regions
q_{smin}	0.2	Minimum quality of q_{sun}
q_{shmin}	0.3	Minimum quality of q_{shadow}

Table E.5: Parameters used in the experiments. The parameters below the line apply only for the proposed method. Adapted from [2].

result of two parallel pipelines. One important characteristic of fusion algorithms is neglected by this design choice. Fused data or fused results generally differ from the original inputs and may, therefore, contain new features and novel information. A simple selection obviously makes this impossible [2].

The FAR of both the proposed method and the RGBT approach mirror the weaker modality. The reason is that high evidence of foreground objects in one modality may still be present after fusion of the data. Only false positives based on weak evidence are successfully smoothed out. In the worst case, false positives from both modalities are present in the result [2].

A good example of the superiority of fusion approaches in terms of DR is given by the sequence INO TreesAndRunner. Obviously, both fusion approaches, i.e., the proposed method and the RGBT approach, perform much better than the single modalities. This better performance is seen because both RGB and thermal contain frames that are very hard to segment. The runner, for example, will pass trees and other objects. Nevertheless, the fusion approaches can still rely on the second modality when the information content of the first is low [2].

	Proposed	RGB	Thermal	RGBT	Select
Day	**0.99**	0.93	0.95	0.97	0.93
	0.30	**0.09**	0.31	0.29	**0.09**
Night	0.84	0.78	0.48	**0.89**	0.78
	0.31	0.69	0.32	0.66	0.69
Auto Gain	**0.94**	0.86	0.73	0.91	0.81
	0.25	**0.09**	0.76	0.40	0.58
Heavy Rain	**0.92**	0.46	0.69	0.48	0.69
	0.22	0.26	**0.11**	0.27	**0.11**
Snowing	**0.96**	0.79	0.21	0.92	0.21
	0.52	0.52	**0.25**	0.55	**0.25**
INO ParkingEvening	**0.95**	0.93	0.91	**0.95**	0.91
	0.26	0.27	**0.18**	0.29	0.18
INO ParkingSnow	0.98	0.86	**0.99**	0.96	**0.99**
	0.32	0.78	0.40	0.35	0.40
INO CoatDeposit	**0.97**	0.10	0.10	0.10	0.10
	0.19	**0.12**	0.30	0.16	**0.12**
INO TreesAndRunner	**0.94**	0.88	0.84	0.93	0.84
	0.44	0.65	**0.36**	0.70	**0.36**
OTCBVS 3	**0.95**	0.75	0.94	0.90	0.78
	0.56	0.96	0.74	0.96	0.93
OTCBVS 4	**1.00**	0.94	0.78	0.99	0.78
	0.55	**0.15**	0.68	0.48	0.68
Average	**0.95**	0.76	0.70	0.83	0.72
	0.35	0.39	0.39	0.46	0.41

Table E.6: Experimental results; first line, Detection Rate (DR), and the second line, False Alarm Rate (FAR). The best DR and FAR values of each set are marked in bold. The proposed method is compared to individual processing of RGB and thermal (RGBT) frames, naive fusion of RGBT frames and "select", which indicates result selection based on quality heuristics [25]. From [2].

7.4 Special Situation Performance

In the following, the results of the specific test sequences are elaborated in detail. It is shown how the performance of the proposed algorithm is affected by different details of the design. Four different problems that tend to arise during outdoor surveillance are discussed. Emphasis has been put on the adaptive modality weighting of the proposed algorithm and its effect on the segmentation results. This context awareness is initially discussed below [2].

Context Awareness

One of the core contributions of this work is the context awareness of the fusion. It is based on a set of quality indicators that have been defined in Section 3. The goal is to evaluate the usefulness of each modality. Instead of using information from the images themselves, contextual information from outside sources has been consulted. Solely the thermal domain has been rated by its own information content. For the tested sequences, the weights calculated by the indicators are more or less fixed. The time frames are simply too short to see an effect based on, for example, the quality indicator covering the altitude of the Sun. The overall concept, however, has been tested by selecting scenes with various conditions [2].

In Figure E.12a,b, quality functions covering a full day are plotted. The plotted day was a summer day with rather good weather. Because of an overcast sky, no cast shadows were detected for this particular morning [2]. Around noon time, the temperature was so high that the thermal camera was overexposed. Figure E.12c shows the entropy-based quality indicator for a full day, which includes the 'snowing' sequence. The sequence starts at 13:17, and one may identify sharp drops in the quality indicators related to the snowfall.

Parameter Sensitivity

The experimental results have been obtained using the quality indicators listed in Table E.4. In the following, we will perform a sensitivity analysis to judge the effect of changing the parameters that guide the context-aware fusion. We use the Snowing and INO ParkingEvening scenes and vary the parameters q_{sun} and q_{weather} in the interval $[0.1, 1.0]$. When both q_{sun} and q_{weather} are low, the resulting weight of the RGB image, w_{RGB}, will be low and the distance map of the thermal image will dominate the fusion process. When both indicators are set to one, the resulting weights will depend on the thermal entropy and the unpredictable quality indicators for both modalities. However, the resulting value of w_{RGB} will be relatively higher.

The DR and FAR rates of the two scenes for varying values of q_{sun} and q_{weather} are shown in Figure E.13. The values used by the general experiments listed in Table E.6 are enclosed by a rectangle.

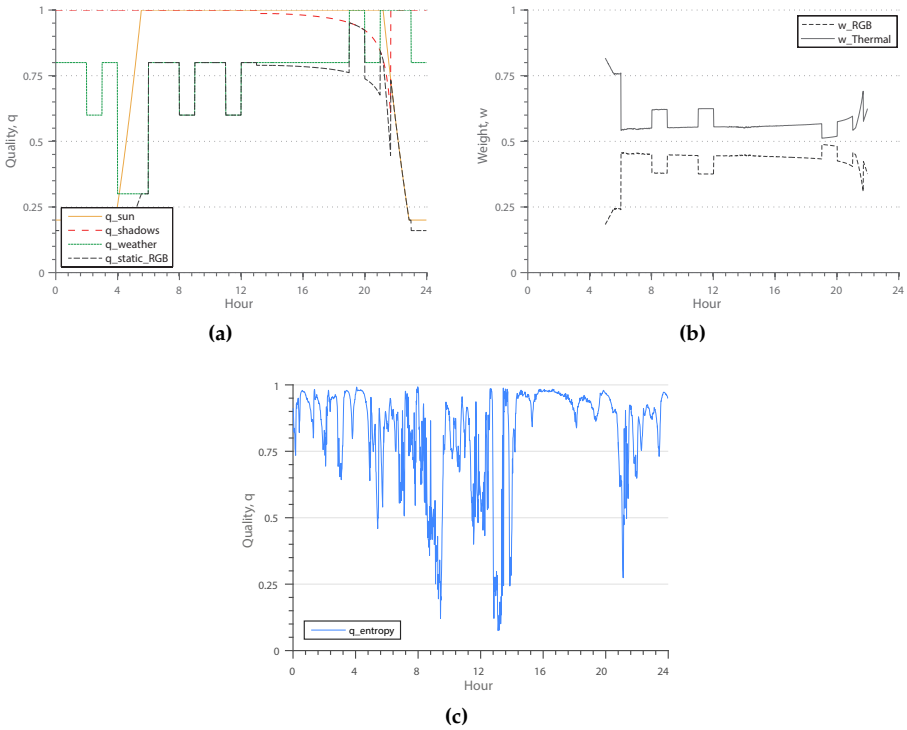


Fig. E.12: RGB and thermal quality indicators and resulting weights of a full day. (**a**,**b**) have been computed on the same sequence, whereas (**c**) shows the thermal quality indicator for the day that includes the "snowing" sequence. The snowfall starts at 13:17 and severely effects the quality of the thermal image. (**a**) Predictable RGB quality indicators and resulting RGB quality, $q_{\text{static}_{\text{RGB}}}$, over a full day; (**b**) weights of the RGB and thermal modalities, w_{RGB} and w_{Thermal}, over a full day; (**c**) entropy-based quality indicator, q_{entropy}, for the thermal domain. (**a**),(**b**) from [2].

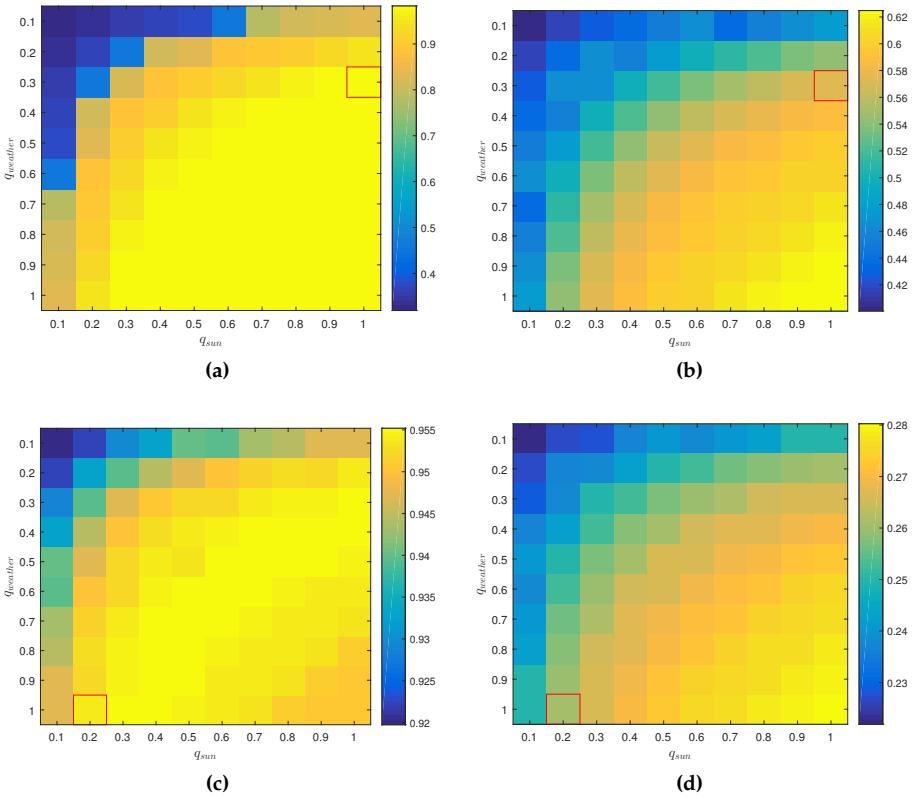


Fig. E.13: Experimental results on the Snowing and INO ParkingEvening sequences. DR and FAR are shown with varying values of the quality indicators q_{sun} and q_{weather}. (**a**) DR, Snowing; (**b**) FAR, Snowing; (**c**) DR, INO ParkingEvening; (**d**) FAR, INO ParkingEvening.

The figures reveal differences on the reliance on the RGB and thermal modalities for the two scenes. The Snowing scene is more reliant on the RGB image than INO ParkingEvening, which shows comparatively little improvement in DR rates when integrating the RGB image into the fusion. By setting $q_{\text{sun}} = q_{\text{weather}} = 0.1$, the Snowing sequence returns a DR below 0.4, whereas the INO ParkingEvening sequence holds a relatively high DR of 0.92. In general, the results show the importance of integrating context-aware quality indicators; a naive fusion, as exemplified by $q_{\text{sun}} = q_{\text{weather}} = 1$, does not always give the best compromise of DR and FAR.

Automatic Gain Control

When large or hot objects enter the scenery, the camera automatically adjusts its gain in order to preserve a high level of detail. This behavior, however, seriously disturbs the

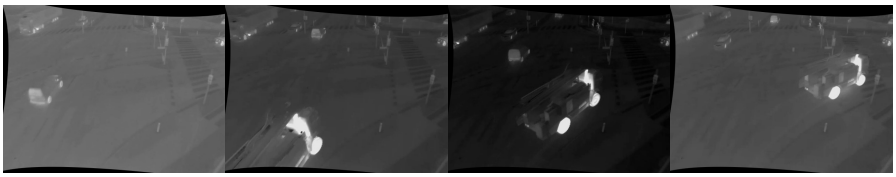


Fig. E.14: Automatic Gain Control (AGC) of the IR camera triggered by a big truck coming into the scenery. (**a**) Frame 170; (**b**) Frame 200; (**c**) Frame 230; (**d**) Frame 260. From [2].

segmentation algorithm, which results in a high number of false positive foreground pixels. Figure E.14 displays the described phenomena. The main challenge is thus the short time frame for adjustment. When the objects leave the scene, the camera re-adjusts the gain. Therefore, the problem often persists only for 100–200 frames, and yet, it highly affects the segmentation results [2].

As seen in Table E.6, the proposed algorithm handles the described problem well. No segmentation quality reduction can be detected from the raw numbers. The reason for this is the adaptive weighting performed in the fusion step. Through the foreground ratio evaluation, which was described in Section 3.3, it can be detected that the background model of the thermal domain is invalid. As a result, the thermal weight function drops to zero, and the segmentation relies on the RGB domain only. Figure E.15 displays this behavior. It can clearly be seen that the quality function drops parallel to the weight of the thermal domain. When the truck has left the scene, the camera adjusts back to normal, and the quality function instantly rises. The weight, however, increases only gradually. This delay is necessary in order to give the background model sufficient time to relearn the background model [2].

Changing Illumination

A very similar problem, which is commonly seen in outdoor surveillance, is changing illumination conditions. The segmentation algorithm is designed to adapt only to slow changes, e.g., shadows moving during daytime, whereas fast changes in the scenery will cause false detection of foreground objects.

Similar to the problem of automatic gain control, which was discussed above, the foreground ratio of the RGB domain will rise because the background model does not adapt fast enough for these changes. Consequently, a weight shift to the thermal domain will be performed by the algorithm, which contributes to the comparatively low FAR of 0.56 in the OTCBVS 3 sequence.

Artifact Reduction

Another contribution of the proposed algorithm can be seen in the results of OTCBVS 3. With 0.56*, the FAR is even lower than the results found for the thermal background*

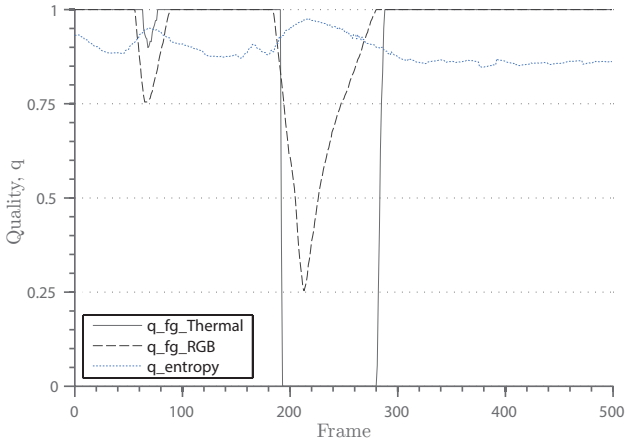


Fig. E.15: The entropy quality indicator, q_{entropy}, and the quality indicators of the unpredictable conditions, $q_{\text{fg}_{\text{RGB}}}$ and $q_{\text{fg}_{\text{Thermal}}}$, for the Auto Gain test sequence. The $q_{\text{fg}_{\text{Thermal}}}$ drops rapidly when a truck enters the scene around Frame 230. The corresponding frames are seen from Figure E.14.

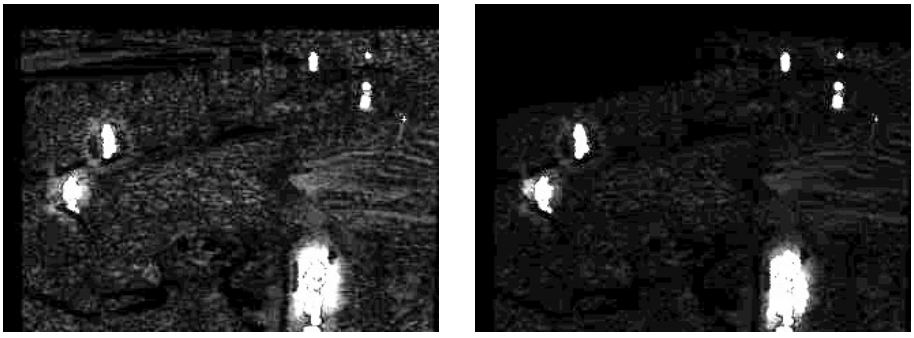
subtraction. This can be reasoned with the adjustments made to the fused distance map on the basis of the scene geometry. Artifacts are unlikely to appear since unpredicted foreground regions are reduced in the distance map. The effect on the foreground mask can clearly be seen in Figure E.16, particularly when compared to the approach using solely the thermal domain [2].

Long-Staying Objects

The GMM background subtraction presented by Stauffer and Grimson [28] assumes that foreground objects are constantly in motion, but this is obviously not the case for all traffic. This issue has been addressed by Yao and Ling [31], and the proposed method has been integrated in our work. The original algorithm causes long-staying objects to gradually merge into the background. This problem is very visible in the INO CoatDeposit test set. The car entering the scene merges into the background within a few frames as seen in Figure E.17. This merging is stopped by prediction of foreground regions and lowering of their update speed, resulting in a significantly better DR of the proposed method [2].

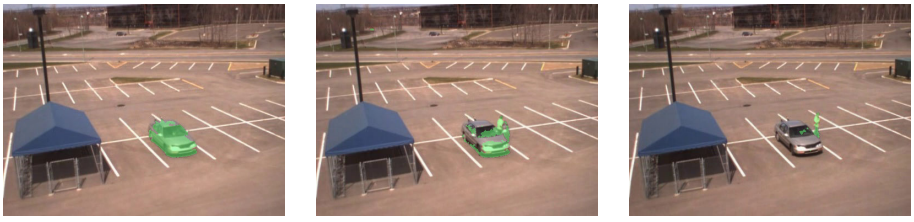
8 Conclusions and Future Perspectives

This paper presented a new approach to multi-modal image fusion. The proposed algorithm fuses the soft segmentation results of two parallel segmentation pipelines based on the RGB and thermal video streams. The fusion is guided by quality indicators for each modality. The quality indicators are



(a) **(b)**

Fig. E.16: Distance map before (**a**) and after scene geometry-based modulation (**b**). From [2].



(a) **(b)** **(c)**

Fig. E.17: The parked car gradually merges into the background by using a standard GMM segmentation method (foreground marked green). (**a**) Car arrives; (**b**) Car has stopped for several frames; (**c**) Driver got out the car. The proposed method mitigates this issue by predicting foreground regions and lowers the update rate α correspondingly. From [2].

based on both image structure and external sources of information. These include the entropy of the thermal image, the altitude of the Sun, the weather conditions of the scene and rapid changes in the resulting output of the parallel segmentation pipelines. *To match the requirements derived from the purpose of traffic monitoring, extensions to the core contribution have been introduced* [2].

The proposed method has been thoroughly tested [2]. The results show that the proposed method performs significantly better than naive fusion of both modalities and consistently better than utilizing a single modality alone. The evaluated performance suggests that the strategy of including image quality indicators in the segmentation process has great potential in future applications.

A common problem of image fusion techniques can be seen from the experimental results. Although the algorithm features a suppression of false positives, a propagation to the fused mask can still be noticed. This is especially the case when the quality rating of the two modalities is similar and information therefore fuses in equal proportions. Based on this observation, further development of the proposed method can be derived. Serrano-Cuerda et al. [25] perform a switch based on image quality indicators, whereas this work performs an adaptive fusion. The next logical step would be to perform the fusion adaptive per image region. By specifying quality indicators for image samples, information about shadows and different lighting conditions within the scene could be considered [2].

The work has been limited to the usage of RGB and thermal imagery. However, the algorithm can easily be adapted to work with different imaging sensors. A setup of the proposed system in combination with sensors helping to estimate the image quality would also be an interesting extension. Weather stations and street temperature sensors would enable the indicators to work much more accurately [2].

Acknowledgments

The recording of the video data was funded by the the Danish Road Directorate. The research has received funding from the European Union's Framework Programme for Research and Innovation, Horizon 2020, under Grant Agreement 635895.

Thiemo Alldieck conceived and designed the experiments. Thiemo Alldieck and Chris H. Bahnsen performed the experiments and analyzed the data. Thiemo Alldieck and Chris H. Bahnsen wrote the paper. Thomas B. Moeslund supervised the entire process.

The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; nor in the decision to publish the results.

References

[1] M. Alemán-Flores, L. Alvarez, L. Gomez, and D. Santana-Cedrés, "Line detection in images showing significant lens distortion and application to distortion correction," *Pattern Recognition Letters*, vol. 36, pp. 261–271, 2014.

[2] T. A. Alldieck, "Information based multimodal background subtraction for traffic monitoring applications," *Unpublished Master Thesis, Aalborg University, Aalborg, Denmark*, 2015.

[3] N. Buch, S. A. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 3, pp. 920–939, 2011.

[4] S. Chen and H. Leung, "An em-ci based approach to fusion of ir and visual images," in *Information Fusion, 2009. FUSION'09. 12th International Conference on*. IEEE, 2009, pp. 1325–1330.

[5] T.-H. Chen, J.-L. Chen, C.-H. Chen, and C.-M. Chang, "Vehicle detection and counting by using headlight information in the dark environment," in *Intelligent Information Hiding and Multimedia Signal Processing, 2007. IIHMSP 2007. Third International Conference on*, vol. 2. IEEE, 2007, pp. 519–522.

[6] C. O. Conaire, N. E. O'Connor, E. Cooke, and A. F. Smeaton, "Comparison of fusion methods for thermo-visual surveillance tracking." in *FUSION*, 2006, pp. 1–7.

[7] J. W. Davis and V. Sharma, "Background-subtraction using contour-based fusion of thermal and visible imagery," *Computer Vision and Image Understanding*, vol. 106, no. 2, pp. 162–182, 2007.

[8] R. Gade and T. B. Moeslund, "Thermal cameras and applications: a survey," *Machine Vision and Applications*, vol. 25, no. 1, pp. 245–262, 2014.

[9] K. Garg and S. K. Nayar, "Vision and rain," *International Journal of Computer Vision*, vol. 75, no. 1, pp. 3–27, 2007.

[10] D. Hall and J. Llinas, *Multisensor data fusion*. CRC press, 2001.

[11] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[12] J. P. Heather and M. I. Smith, "Multimodal image registration with applications to image fusion," in *Information Fusion, 2005 8th International Conference on*, vol. 1. IEEE, 2005, pp. 8–pp.

[13] T. Hrkać, Z. Kalafatić, and J. Krapac, "Infrared-visual image registration based on corners and hausdorff distance," in *Image Analysis*. Springer, 2007, pp. 383–392.

[14] R. Istenic, D. Heric, S. Ribaric, and D. Zazula, "Thermal and visual image registration in hough parameter space," in *Systems, Signals and Image Processing, 2007 and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services. 14th International Workshop on*. IEEE, 2007, pp. 106–109.

[15] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis, "A survey of video processing techniques for traffic applications," *Image and Vision Computing*, vol. 21, no. 4, pp. 359–381, 2003.

[16] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground–background segmentation using codebook model," *Real-time imaging*, vol. 11, no. 3, pp. 172–185, 2005.

[17] H. Kwon, S. Z. Der, and N. M. Nasrabadi, "Adaptive multisensor target detection using feature-based fusion," *Optical Engineering*, vol. 41, no. 1, pp. 69–80, 2002.

[18] E. Lallier and M. Farooq, "A real time pixel-level based image fusion via adaptive weight averaging," in *Information Fusion, 2000. FUSION 2000. Proceedings of the Third International Conference on*, vol. 2. IEEE, 2000, pp. WEC3–3.

[19] J. J. Michalsky, "The astronomical almanac's algorithm for approximate solar position (1950–2050)," *Solar energy*, vol. 40, no. 3, pp. 227–235, 1988.

[20] National Weather Service, National Oceanic and Atmospheric Administration's, "Weather Element List and Suggested Icons," http://w1.weather.gov/xml/current_obs/weather.php, 2016, [Online; accessed 28-September-2016].

[21] M. Nieto, L. Unzueta, J. Barandiaran, A. Cortés, O. Otaegui, and P. Sánchez, "Vehicle tracking and classification in challenging scenarios via slice sampling," *EURASIP Journal on Advances in Signal Processing*, vol. 2011, no. 1, pp. 1–17, 2011.

[22] A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara, "Detecting moving shadows: algorithms and evaluation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 7, pp. 918–923, 2003.

[23] I. Ridpath, *A dictionary of astronomy*, 2nd ed. Oxford University Press, 2012.

[24] K. Robert, "Night-time traffic surveillance: A robust framework for multi-vehicle detection, classification and tracking," in *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*. IEEE, 2009, pp. 1–6.

[25] J. Serrano-Cuerda, A. Fernández-Caballero, and M. T. López, "Selection of a visible-light vs. thermal infrared sensor in dynamic environments based on confidence measures," *Applied Sciences*, vol. 4, no. 3, pp. 331–350, 2014.

[26] P. Shah, S. Merchant, and U. B. Desai, "Fusion of surveillance images in infrared and visible band using curvelet, wavelet and wavelet packet transform," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 8, no. 02, pp. 271–292, 2010.

[27] L. St-Laurent, X. Maldague, and D. Prévost, "Combination of colour and thermal sensors for enhanced object detection," in *Information Fusion, 2007 10th International Conference on*. IEEE, 2007, pp. 1–8.

[28] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2. IEEE, 1999.

[29] E. Strigel, D. Meissner, and K. Dietmayer, "Vehicle detection and tracking at intersections by fusing multiple camera views," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 882–887.

[30] M. Vollmer and K.-P. Möllmann, *Infrared thermal imaging: fundamentals, research and applications*. John Wiley & Sons, 2010.

[31] L. Yao and M. Ling, "An improved mixture-of-gaussians background model with frame difference and blob tracking in video stream," *The Scientific World Journal*, vol. 2014, 2014.

[32] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2. IEEE, 2004, pp. 28–31.

[33] Q. Zou, H. Ling, S. Luo, Y. Huang, and M. Tian, "Robust nighttime vehicle detection by tracking and grouping headlights," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 16, no. 5, pp. 2838–2849, 2015.

[34] Y. Zou, G. Shi, H. Shi, and Y. Wang, "Image sequences based traffic incident detection for signaled intersections using hmm," in *Hybrid Intelligent Systems, 2009. HIS'09. Ninth International Conference on*, vol. 1. IEEE, 2009, pp. 257–261.

Part III

Robust Traffic Analysis

Paper F

Rain Removal in Traffic Surveillance: Does it Matter?

Chris H. Bahnsen and Thomas B. Moeslund

The paper has been published in the
IEEE Transactions on Intelligent Transportation Systems (Early Access), 2018,
pp. 1–18.

The layout has been revised.

Abstract

Varying weather conditions, including rainfall and snowfall, are generally regarded as a challenge for computer vision algorithms. One proposed solution to the challenges induced by rain and snowfall is to artificially remove the rain from images or video using rain removal algorithms. It is the promise of these algorithms that the rain-removed image frames will improve the performance of subsequent segmentation and tracking algorithms. However, rain removal algorithms are typically evaluated on their ability to remove synthetic rain on a small subset of images. Currently, their behavior is unknown on real-world videos when integrated with a typical computer vision pipeline. In this work, we review the existing rain removal algorithms and propose a new dataset that consists of 22 traffic surveillance sequences under a broad variety of weather conditions that all include either rain or snowfall. We propose a new evaluation protocol that evaluates the rain removal algorithms on their ability to improve the performance of subsequent segmentation, instance segmentation, and feature tracking algorithms under rain and snow. If successful, the de-rained frames of a rain removal algorithm should improve segmentation performance and increase the number of accurately tracked features. The results show that a recent single-frame based rain removal algorithm increases the segmentation performance by 19.7% on our proposed dataset, but it eventually decreases the feature tracking performance and showed mixed results with recent instance segmentation methods. However, the best video-based rain removal algorithm improves the feature tracking accuracy by 7.72%.

1 Introduction

Monitoring of road traffic is usually performed manually by human operators who observe multiple video streams simultaneously. However, the manual monitoring is both tiresome and does not scale with the growing number of cameras and an increased appetite for a deeper understanding of road user behavior. Thus, there is a clear-cut case for computer vision methods to step in and automate the process. If successful, vision methods in road user detection, classification, and tracking could give valuable insights into and analysis of road user behavior and accident causation, which could ultimately help reduce the number of accidents.

However, most computer vision systems are designed to work under optimal conditions such as clear skies, low reflections, and few occlusions. Whenever one of these constraints is violated, the performance of the vision system rapidly deteriorates, and so does the promise of automated traffic analysis.

Non-optimal conditions are caused by several phenomena, but most prominently by bad weather conditions. We may divide bad weather into two main groups: steady and dynamic conditions [24]. Steady weather conditions in-

clude fog, mist, and haze, which degrade the contrast and reduce the visibility of the scene. Dynamic weather conditions include rainfall and snowfall, which appear as spatio-temporal streaks in the surveillance video, which may temporarily occlude objects with close proximity to the camera. Objects at greater distance from the camera are affected by the accumulation of rain and snow streaks, which reduces the visibility of the scene much like fog, mist, and haze.

We differentiate between three different approaches to cope with the challenges of bad weather in automated video surveillance: to mitigate the effects by pre-processing the video, to strengthen the robustness of the core vision algorithms, or to augment the sensing system by the use of multiple multi-modal sensors.

In this work, we will study the implications of pre-processing the input video signal by algorithms that mitigate the dynamic effects of rainfall and snowfall. Many authors of such rain or snow removal algorithms note that these algorithms could help improve the robustness of traditional vision methods such as segmentation, classification, and tracking. We will investigate this claim through quantitative analysis and hereby provide valuable insights into this field for the benefit of the entire research community.

Current evaluations of rain removal algorithms are based on short video sequences or a collection of still images, typically provided by the authors themselves. Quantitative results are obtained by removing rain on synthetic datasets, where rain streaks are overlaid on rain-free images [75]. Is is common to see indoor images with synthetic rain as part of training [79] and testing [81] of rain removal algorithms. The performance of rain removal algorithms is usually measured by calculating the Structural Similarity Index (SSIM) [71] and the Peak Signal-to-Noise-Ratio (PSNR) between the de-rained and the rain-free images. However, a good SSIM or PSNR score on a synthetic dataset does not necessarily translate into performance when the rain removal algorithms are used on real-world footage. Such evaluation is typically performed by inspection of a limited selection of real-world rainy images.

We are curious how rain removal algorithms will work on traffic surveillance video under real-world conditions that include rainfall and snowfall, and how they affect a traditional computer vision pipeline. In this work, we want to measure the effectiveness of a rain removal algorithm, not by using the raw properties of the produced rain-removed images but by using the performance of the subsequent segmentation, instance segmentation, and feature tracking algorithms that run on top of the rain-removed imagery. If effective, a rain removal algorithm should improve the performance of the subsequent algorithms.

Our contributions are the following:

1. We provide a comprehensive overview of rain removal algorithms, using

both single-image and video-based algorithms.

2. We provide a new publicly available dataset of 22 real-world sequences from 7 urban intersections in various degrees of bad weather involving rain or snowfall. Each sequence has a duration of 4-5 minutes and is recorded with both a color camera and a thermal camera.

3. We use this dataset and the BadWeather training sequences of the Change Detection 2014 challenge [69] to assess the performance of classic segmentation methods and recent instance segmentation methods on the raw and rain-removed imagery. Furthermore, we use the forward-backward feature tracking accuracy to investigate if feature-based methods perform better under rain-removed imagery.

4. The entire evaluation protocol and our implementation of the rain removal algorithm of Garg and Nayar [24] is publicly available as open-source[1,2] to enable others to build upon our results.

The rest of the paper is organized as follows: Section 2 describes how rain and snowfall impair the visual surveillance footage in traffic scenes. Section 3 gives a comprehensive overview of rain removal algorithms and their general characteristics. Our new dataset is presented in Section 4. The evaluation protocol of selected rain removal algorithms on this dataset is presented in Section 5, and the results hereof are treated in Section 6. Section 7 concludes our work.

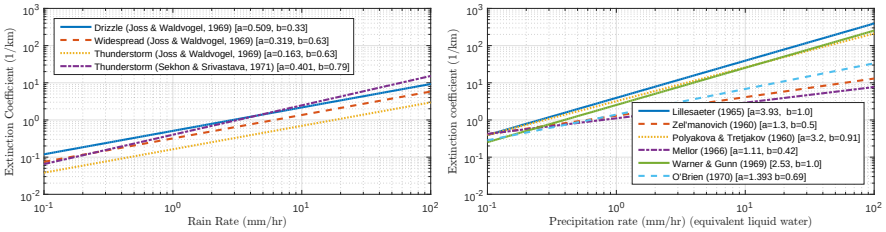
2 The Impact of Rain and Snow

Bad weather, including rain and snow, is generally acknowledged as a challenge in computer vision [9], but little work has been undertaken to identify the severity of this problem. In this section, we will shed light on the impact of rain and snow in traffic surveillance, and how it might affect the vision systems that are built on top of the video streams.

Rainfall and snowfall will have a negative effect on the visibility of the scene due to the atmospheric scattering and absorption from raindrops and snowflakes. In the atmospheric sciences, the combined impact of scattering and absorption from a particle is called extinction [64]. For wavelengths in the visible and infrared range, the extinction from raindrops can be approximated as [58]:

$$\beta_{\text{ext}_{\text{rain}}} = A \cdot R^B \tag{F.1}$$

[1] `https://bitbucket.org/aauvap/aau-rainsnow-eval`
[2] `https://bitbucket.org/aauvap/rainremoval/`



(a) Extinction during rainfall **(b)** Extinction during snowfall

Fig. F.1: Degradation of visibility due to extinction from rainfall and snowfall. Parameters as reported in [48, 58].

where $\beta_{\text{ext}_{\text{rain}}}$ is the rain extinction coefficient, A and B are model parameters, and R is the rain rate in mm/hr.

The exact values of the parameters A and B vary according to the precipitation type. Shettle [58] reports five different models for rain. We leave out the oldest rain model and plot the remaining four in Figure F.1a. The approximation used in Equation F.1 builds on the notion that the physical size of the raindrops is much greater than the wavelength in consideration. For radiation of longer wavelengths, a wavelength-dependent correction must be added [58].

The extinction caused by falling snowflakes in the visible range uses a model similar to Equation F.1. However, one must convert the snow depth to equivalent liquid water. For wet snow, 1 mm of snow corresponds to 5 mm of rain, whereas for dry snow, the conversion range is greater than 1 to 20. For our calculations, we use the 'rule-of-thumb' approximation by Shettle [58] with a ratio of 1 to 10. When converted, the model is defined as:

$$\beta_{\text{ext}_{\text{snow}}} = A(1/10)^B \cdot S^B \tag{F.2}$$

where $\beta_{\text{ext}_{\text{rain}}}$ is the snow extinction coefficient, S is the rate of snow accumulation, and other parameters are defined in Equation F.1.

In Figure F.1b, we plot the six different sets of model parameters reported by Mason [48]. Depending on the chosen model, one can see that the extinction from snow is a half-magnitude greater than the equivalent amount of rainfall. Thus, the visibility of a scene is reduced more under snow than under rain.

The properties of rainfall as they are observed by a typical surveillance camera have been studied extensively by K. Garg and S.K. Nayar, who, in their seminal work [24], laid out the theoretical framework for the physical and practical implications of rain in vision. They provided a physical model of a falling raindrop and constructed an appearance model for the raindrop as viewed by a camera. We will summarize the relevant findings of this work below:

Table F.1: The Visual Effects of Rain and Snow

Phenomena	Visual effect
Raindrop	Spatio-temporal streaks, duration approx. 1 frame per streak
Snowflake	Spatio-temporal streaks, duration approx. 1 frame per streak
Dense rain and snow	Reduced visibility, depth of field
Raindrops on lens	Blur, diffuse scattering of light
Puddles	Surroundings reflected by puddles and splashes from road users

1. Raindrops are transparent, and most drops are less than 1 mm in size.
2. The motion of a raindrop can be modeled as a straight line.
3. A raindrop appears brighter than its background. The change in intensities caused by a falling rain streak is linearly related to the background intensities that are occluded by the rain streak.

These observations, especially the notion that raindrops are brighter than the background, have had a major impact on all subsequent works on video-based rain removal. We will revisit these observations in our survey of rain removal algorithms in Section 3. The implications of rain and snow on visual surveillance are not limited to the characteristics of a raindrop alone. The accumulation of rain on surfaces eventually leads to puddles when the drainage of the road is insufficient. When vehicles or other road users drive through these puddles, water will splash from the wheels. Raindrops may also attach to the lens or even freeze to ice if the camera is not installed inside a protective outdoor housing or the wind is too strong.

Table F.1 provides an overview of how these phenomena affect the observed images in a surveillance setting, whereas Figure F.2 shows examples of footage impaired by snow, heavy rain, raindrops on the lens, and reflections on the road.

It is apparent from Figure F.2 that these phenomena degrade the visibility of the road users as observed by the human eye. The degradation on visibility will inevitably affect vision algorithms due to a reduced signal-to-noise ratio. A detailed treatment on how vision-based segmentation and tracking algorithms are affected by the effects of rain and snow is given in Table F.2.

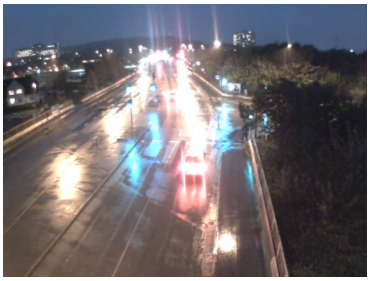
The concrete effects on a state-of-the-art unsupervised segmentation algorithm [60] is shown in Figure F.3.

We may also infer the impact of rain and snow from the results of existing challenges and datasets. The most prominent dataset on background segmentation, ChangeDetection.net [69], features a 'BadWeather' category that

(a) Snow

(b) Heavy rain

(c) Raindrops on the lens



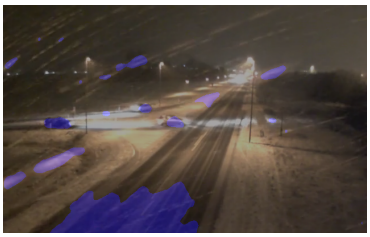
(d) Reflections

Fig. F.2: Visual examples of rain and snow in traffic surveillance. In (a) and (d), bad lighting conditions further deteriorate the visibility of the scene.

Table F.2: The Effects of Rain and Snow on Segmentation and Tracking

Phenomena	Segmentation	Tracking
Raindrop	Non-constant background, false detections	False features, disturbances in tracking
Snowflake	Non-constant background, false detections	False features, disturbances in tracking
Dense rain and snow	Missing detections of far-away objects	Fewer salient features of far-away objects
Raindrops on lens	Missed detections in area of raindrop	No features in area of raindrop
Puddles	False detections due to reflections	False features due to reflections

contains a total of 20,900 video frames distributed in four different scenes. Snow and snowfall are the common denominators for the 'BadWeather' sequences, but the exact nature of the scenes varies. In Table F.3, we have summarized the latest results of the ChangeDetection.net challenge for the BadWeather sequence, a trivial 'Baseline' sequence, and a weighted average



(a) Snow

(b) Heavy rain

(c) Raindrops on the lens

(d) Reflections

Fig. F.3: Segmentation results of a state-of-the-art segmentation algorithm [60] on the sequences shown in Figure F.2. The segmented masks are overlaid in blue. In (a) and (b), some snowflakes or rain streaks are detected as foreground. In (c), the raindrops on the lens lead to missing detections of parts of the red vehicle. In (d), reflections from the tail lights are detected as foreground, whereas most of the car is not.

Table F.3: Average F-measure of the Best Segmentation Algorithm on the ChangeDetection.net [69] Database

Method	BadWeather	Baseline	Overall
Best supervised	0.98	0.99	0.97
Best unsupervised	0.87	0.96	0.79

of all sequences in the database. We distinguish between supervised change detection methods, which use the training samples of each sequence, and unsupervised change detection methods, which use default parameters for all sequences.

The comparison between the BadWeather and Baseline sequences suggests that especially unsupervised change detection methods face difficulties when the video sequences are captured under bad weather conditions. However, the overall performance of the entire database is even lower than the BadWeather sequences. The best results on nighttime videos and the dynamic pan-tilt camera footage are dramatically lower with best f-measures of 0.76 for unsupervised change detection methods [69]. This suggests that a combination of rain or snow on nighttime videos may be increasingly difficult. It should

be noted, however, that an extensive comparison of weather phenomena in surveillance requires that one can change only one parameter at the time, which is hardly the case with real-life surveillance footage.

3 Rain Removal Algorithms

The work within rain removal may be divided into two main categories: video-based rain removal, where the temporal information of a video stream is used to detect and remove the rain streaks, and single-image based rain removal, where such temporal information is not provided or used. We also make the distinction between rain streaks and rain drops. A rain streak is defined as a spatio-temporal effect with the approximate duration of one frame, whereas a rain drop is attached to the lens of the camera and remains stationary for seconds or even minutes. The published work on rain drop removal is significantly smaller than the corresponding work on rain streak removal. When temporal information is available, we consider rain drop removal to be considerably easier than rain streak removal. In this work, we will focus on the removal of rain streaks. However, the detection and removal of rain drops could be added as an additional pre-processing step, either before or after rain streak removal. Notable efforts on rain drop removal include the work of [16], [76], and [53].

Rain removal can also be seen as a special case of image denoising. A good overview of general image denoising techniques is given in [55]. Image dehazing [37] and defogging [73] are also closely related fields that mitigate the effects of steady bad weather conditions. An overview of rain streak removal techniques is available in Table F.4. In the overview table, we note if the high-frequency (HF) parts of the image or video are explicitly computed as part of the rain streak removal. If so, we note the name of the used filter. Furthermore, we categorize the algorithms on the basis of how they learn from data and use the following criteria:

Manual: Requires manual hand-tuning of the algorithm for each particular sequence.

Fixed: The algorithm does not contain any adjustable parameters or parameters are provided 'as-is' by the original authors. Parameter tuning by the original authors is performed by hand and is based on empirical observations.

Online: One or more parameters are learned from the current input image or video. Contains no offline learning.

Offline: One or more parameters are learned from an offline database. The algorithm may also contain online learning.

The notion of snow removal is tightly coupled with the work within rain streak removal. In fact, the earliest rain removal technique of Table F.4 deals with noise elimination from snowfall [25]. Most authors of rain streak removal algorithms evaluate their rain removal algorithms solely on images of rainfall, but some also include images or videos of snowfall [5,6,36]. As summarized in Table F.1, the impact of rain and snow is similar in the visible spectrum, so it is natural to make a joint study of the two phenomenons. In the remainder of this article, we will refer to rain streak and snow removal jointly as rain removal.

Table F.4: Rain Removal Algorithms. We list both single-frame (image) and video-based methods and their main method for rain streak detection. Manual and fixed learning indicate hand-tuning by the original authors, whereas online and offline learning learns from the input image, input video, or a collection of offline images. If the method computes high-frequency images, we note the name of the filter.

Paper	Year	Author group	Input	Color space	Main detection method	Learning	High-frequency image
[79]	2017	1	Image	RGB	Conditional generative adversarial network	Offline	No
[78]	2017	1	Image	RGB	Sparse dictionary	Offline	No
[19]	2017	2	Image	RGB	Convolutional neural network	Offline	Guided
[20]	2017	2	Image	RGB	Convolutional neural network	Offline	Guided
[32]	2017	None	Video	RGB	Matrix decomposition, orientation	Online	No
[52]	2017	None	Video	RGB	Matrix decomposition, intensity	Online	No
[75]	2017	None	Image	RGB	Convolutional neural network	Offline	No
[44]	2017	None	Image	RGB	Convolutional neural network	Offline	No
[10]	2017	None	Image	Grey	Sparse dictionary	Offline	No
[70]	2017	None	Image	RGB	Sparse dictionary, orientation, color	Offline	Guided
[39]	2016	None	Image	YUV	Matrix decomposition, dictionary	Offline	No
[36]	2015	3	Video	RGB	Sparse dictionary	Offline	No
[45]	2015	None	Image	RGB	Matrix decomposition, sparse dictionary	Online	No
[54]	2015	None	Video	RGB	Photometric, chromatic constraint, orientation	Fixed	No
[11]	2014	4	Image	RGB	Sparse dictionary	Online	Guided
[29]	2014	4	Image	RGB	Sparse dictionary	Online	Bilateral
[62]	2014	None	Image	RGB	Sparse dictionary	Online	Guided
[51]	2014	None	Image	HSV	Color thresholding, orientation	Fixed	No
[1]	2014	None	Video	RGB	Matrix decomposition	Online	No
[12]	2014	None	Video	RGB	Photometric constraint, optical flow	Fixed	No
[35]	2013	3	Image	RGB	Streak orientation, intensity	Fixed	No
[67]	2013	5	Video	YCbCr	Photometric constraint, orientation	Fixed	No
[14]	2013	None	Video	RGB	Matrix decomposition	Online	No
[81]	2013	None	Image	RGB	Filtering	Fixed	Guided
[30]	2012	4	Video	RGB	Sparse dictionary	Online	Bilateral
[34]	2012	4	Image	RGB	Sparse dictionary	Online	Bilateral

Continued on next page

Table F.4 – continued from previous page

Paper	Year	Author group	Input	Color space	Main detection method	Learning	High-frequency image
[72]	2012	6	Image	YUV	Filtering	Fixed	No
[74]	2012	None	Video	RGB	Photometric constraint	Fixed	Bilateral
[68]	2012	5	Video	YCbCr	Photometric constraint	Fixed	No
[21]	2011	4	Image	RGB	Sparse dictionary	Online	Bilateral
[66]	2011	5	Video	RGB	Photometric constraint, orientation	Fixed	No
[57]	2011	None	Video	RGB	Photometric constraint	Fixed	No
[6]	2011	None	Video	RGB	Background segmentation, photometric constraint, orientation	Online	No
[82]	2011	None	Video	RGB	Filtering, chromatic constraint, streak intensity	Fixed	No
[5]	2010	None	Video	RGB	Frequency space, streak orientation, intensity	Manual	No
[43]	2009	6	Video	RGB	Photometric, chromatic constraint	Fixed	No
[8]	2008	None	Video	RGB	Photometric constraint, streak orientation	Fixed	No
[50]	2008	None	Video	RGB	Photometric constraint, streak orientation	Fixed	No
[24]	2007	None	Video	RGB	Photometric constraint, streak orientation, size	Fixed	No
[80]	2006	None	Video	RGB	Streak intensity, chromatic constraint	Online	No
[61]	2003	None	Video	RGB	Temporal median	Fixed	No
[25]	1999	None	Video	RGB	Temporal median	Fixed	No

3.1 Single-Image Based Rain Removal

Rain removal from a single image is hard. Rain is a spatio-temporal phenomena and, without temporal information, one must make an informed guess on the temporal effects. Successful single-image based rain removal algorithms must effectively model the spatial influences of the rain streak and compensate accordingly.

We divide the published methods into four categories: filtering, matrix decomposition, dictionary learning, and convolutional neural networks.

Filtering

Applying one or more filters is a straightforward method to reduce the amount of rain in the image. A guided filter [27] is used in [72] to suppress the rain, where the minimum and maximum RGB values are used as the guidance image of the filter. In the work of Zheng et al. [81], the guided filter is used to split the image into low-frequency (LF) and high-frequency (HF) parts. The pixel-wise minimum of the LF image and of the input image is used as the input of an additional guided filter, which produces the rain-removed image.

Although effective in suppressing the rain, the filter-based methods will also effectively blur textured and detailed parts of the image. In order to improve this, one needs to model the rain streaks.

Matrix Decomposition

A rain streak model is obtained in matrix decomposition techniques by adding additional constraints on the removal process. The rain removal problem is formalized as an exercise in decomposing the input image I into the rain-free image B and the rain image R, such that they add up to comprise the original image:

$$I = B + R \tag{F.3}$$

In order to guide the decomposition, one has to make certain assumptions on the properties of B and R. A common assumption is that B should have low total variation (TV), i.e. the recovered image should be smooth:

$$\mathrm{TV}_B = \sum_{i=1}^{M} \sum_{j=1}^{N} \|\Delta b_{i,j}\|_2 \tag{F.4}$$

where M and N are the dimensions of the image, and $\Delta b_{i,j}$ is the gradient of B at position i, j.

With the exception of dense rain, rain streaks appear relatively infrequently compared to the total number of pixels in an image. Thus, it makes sense to

impose sparsity on R via the squared Frobenius norm [39]:

$$\arg\min_{R} = \|R\|_F^2 = \sum_{i=1}^{M}\sum_{j=1}^{N}|r_{i,j}|^2 \tag{F.5}$$

where $r_{i,j}$ is the pixel value of R at position i, j.

Other methods utilize different norms to induce sparsity. In [78], the 1-norm is used to induce sparsity on B:

$$\arg\min_{B} = \|B\|_1 = \sum_{i=1}^{M}\sum_{j=1}^{N}|b_{i,j}| \tag{F.6}$$

where $b_{i,j}$ is the pixel value of B at position i, j.

Some methods ([1,52]) use the nuclear norm to enforce low rank on B:

$$\arg\min_{B} = \|B_{i,j}\|_* = \mathrm{tr}(\sqrt{B^*B}) \tag{F.7}$$

Solvers Different methods have been proposed to solve the constrained matrix decomposition problem, e.g. the Alternating Direction Method of Multipliers (ADMM) [7], (Robust) Principal Component Analysis (PCA), or the Inexact Augmented Lagrange Multiplier (IALM) [41]. Other solvers are typically used in combination with dictionary learning, which is described in the following. When the decomposition is converging, the background image B is used as the rain-removed image.

The rain removal methods that use a matrix decomposition scheme based on Equation F.3 or variants of it are listed in Table F.5, which shows that the community does not agree on the constraints for R and B. Some works demand sparsity on R [1], while others demand it on B [78]. The same holds for the low rank requirement of R and B. The disagreement sheds light on the general problem of matrix composition techniques: neither the rain nor the background is guaranteed to adhere to the imposed mathematical constraints on the image structure. Thus, there might be high-frequency textures 'trapped' within the segmented rain image and rain streaks 'trapped' within the segmented background image.

Dictionary Learning

Based on the observation that rain streaks fall in the same direction and share similar patterns, one can formulate the rain removal problem as segmenting the image into patches. These patches are classified into rain and non-rain patches by one or more dictionaries. The dictionaries may be learned online from the input image or from a offline bank of (generated) rain streaks.

Table F.5: Rain Removal Algorithms That Include Decomposition or Dictionary Learning

Paper	Year	Author Group	Matrix decomposition	Dictionary learning	Assumptions	Solver or method	Patch-based	High-frequency image
[78]	2017	1	X	X	R low rank, B sparse, B low TV	ADMM	X	No
[32]	2017	None	X		B low temporal variation, B low TV on x, y-axes	ADMM		No
[52]	2017	None	X		B low rank, R, F sparse, R_d Gaussian	MRF, SVD		No
[10]	2017	None		X	R vertical gradients, B high SSIM	OMP, Genetic algorithm	X	No
[70]	2017	None		X	R horizontal gradients, low color, gradient variance	HOG	X	Guided
[39]	2016	None	X	X	R sparse, B low TV, B, R n-modal Gaussian	L-BFGS, GMM	X	No
[36]	2015	3		X	R similar angular components	MCA, SVD, SVM	X	No
[45]	2015	None	X	X	R, B mutually exclusive	OMP, K-SVD	X	No
[11]	2014	4		X	R low depth of field, B chromatic	MCA, HOG	X	Guided
[29]	2014	4		X	R similar gradients	MCA, HOG, PCA	X	Bilateral
[62]	2014	None		X	R high structural similarity	SSIM, K-SVD	X	Guided
[1]	2014	None	X		R sparse, B low rank	PCA		No
[14]	2013	None	X		R linearly dependent, B low TV	IALM	X	No
[30]	2012	4		X	R similar gradients	MCA, HOG, PCA, SVM	X	Bilateral
[34]	2012	4		X	R low gradient variance	MCA, HOG, K-means	X	Bilateral
[21]	2011	4		X	R brighter than neighbors	MCA, K-SVD	X	Bilateral

Dictionaries were introduced in rain removal by Fu et al. in 2011 [21] and improved by Kang et al. in 2012 [34]. The authors applied a variant of the Morphological Component Analysis (MCA) technique [17] on the HF component of the input image to decompose the image into B and R. A sparse coding algorithm [46] is used to learn a dictionary of atoms from patches of the HF image. The Histogram of Oriented Gradients (HOG) is computed for each atom, and the output hereof is used as input to a two-cluster K-means algorithm. The cluster with the smallest gradient variance is selected as the rain atoms. Once the dictionary is classified, Orthogonal Matching Pursuit (OMP) [47] is used to sparsely reconstruct the HF image:

$$\min_{\theta^k_{\text{HF}}} = \|b^k_{\text{HF}} - D_{\text{HF}} \cdot \theta^k_{\text{HF}}\|_2^2 \text{ s.t. } \|\theta^k_{\text{HF}}\|_0 \leq L \tag{F.8}$$

where D_{HF} is the dictionary containing both rain and non-rain atoms of the HF image, b^k_{HF} is the k'th patch of the HF image, θ^k_{HF} is a matrix containing the sparse coefficients for reconstructing the k'th patch, and L is the maximum number of non-zero elements in α. In [34], $L = 10$.

The dictionary components of D_{HF}, which corresponds to the previously classified non-rain atoms, are used to reconstruct the HF part of the rain-removed image. Finally, this image is added to the low frequency (LF) image, and a rain-removed image is obtained. In our experiments, we have experienced that the sparse reconstruction may be entirely composed of what is classified as rain atoms. In this case, the rain-removed image is completely empty.

Huang et al. [30] used the same rain removal framework as [34] but changed the selection of rain and non-rain atoms. Instead of using K-means on the HOG-computed gradient, they used Principal Component Analysis (PCA) to find the dominant directions of intensity change. The most similar and dissimilar atoms are assigned as rain and non-rain atoms, respectively. The remaining atoms are classified by a Support Vector Machine (SVM).

A different variant hereof is presented in [29]. The HOG- and PCA-based approaches are still used to find the discriminative features of the dictionary components, but the grouping of atoms is performed by the use of affinity propagation [18]. A greedy scheme is performed on top hereof to select $K <= 16$ clusters. The variance of the atoms in each cluster is computed, and the cluster with the lowest variance is regarded as the one containing the rain atoms. The remaining clusters are then used to reconstruct the rain-removed image.

Another augmentation of [34] is provided by Chen et al. [11], who introduced a depth of field measure on the atom components. Pixels with a low depth of field are regarded as rain. Furthermore, the work uses chromacity information of the atoms to restore details that might otherwise be regarded as rain atoms. In the independent work of Wang et al. [70], the authors use

the color variance of an atom to refined the HF rain image from [34].

The main shortcoming of the above family of dictionary learning techniques ([11,21,30,34]) is the unilateral dependence on the input image for both the rain and non-rain dictionaries. Even when augmented with additional constraints by Huang [30] and Chen [11], HF details may be integrated into the rain dictionaries and thus be removed from the rain-removed image. In order to combat this problem, it seems that one should resort to offline techniques for dictionary training.

A combination of offline and online training is used by Li et al. [39], who utilize Gaussian Mixture Models (GMM) to learn two dictionaries of rain and non-rain patterns. The non-rain dictionary is learned offline, while the rain streak dictionary is learned using relatively flat regions of the image. The generation of the background and rain image is formulated as an image decomposition problem:

$$\min_{B,R} \|I - B - R\|_F^2 + \alpha\|\Delta B\|_1 - \beta\|R\|_F^2 - \gamma G(B,R) \tag{F.9}$$

where I, B, and R are the input, background, and rain image respectively. α, β, and γ, are scalars estimated heuristically, and $G(B,R)$ is the reconstruction of the background and rain image based on the respective dictionaries.

One sees from Equation F.9 that further constraints are put on the reconstructed background and rain image; the background must have low total variation as defined in Equation F.4, and the rain image must be sparse as defined by the Frobenius norm. The decomposition problem is solved by the L-BFGS algorithm [42].

The rain removal algorithms that use dictionary learning are listed in Table F.5.

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) were introduced to rain streak removal almost simultaneously in [19,20,75,79]. We summarize the work on CNN-based rain removal in Table F.6.

A classical CNN approach was proposed by Fu et al. in [19]. Just like prior work in dictionary learning, the rain removal is performed on the HF components of the input image, which is produced by the guided filter. The network uses three convolutional layers to de-rain the HF image, which is subsequently added to the LF image. Training is performed on synthesized rain images using rain streaks generated in Adobe Photoshop. Subsequent work by the same author [20] used a much deeper network with residual connections [28] and an increased number of training samples (9100).

Yang et al. [75] used dilated convolutions on three different scales [77] to aggregate multi-scale information due to the variable-size receptive field of

Table F.6: Rain Removal Algorithms That Use Convolutional Neural Networks

Paper	Year	Author group	Network type	Convolutional layers	Training size	High-frequency image
[79]	2017	1	Conditional Generative Adversarial Network (Pix2Pix [31])	12 (Generator)	700	No
[19]	2017	2	Convolutional Neural Network	3	4900	Guided
[20]	2017	2	Convolutional Neural Network (ResNet [28])	26	9100	Guided
[75]	2017	None	Convolutional Neural Network (Dilated convolution)	10	< 300	No
[44]	2017	None	Convolutional Neural Network (Inception-V4 [63])	> 76	16.000	No

the dilated convolutions. This helps the network to incorporate contextual information, which might help when learning to remove the rain.

Inspired by the success of generative networks, Zhang et al. [79] used a generative adversarial network (GAN) that is conditioned on the input image. They used the Pix2Pix framework [31] to create a generator network that de-rains an input image whereto artificial rain streaks have been added. Based on appearance, it is the role of an additional discriminator network to judge whether a de-rained image is the output of the generator network or is the original rain-free image.

A dedicated CNN-framework for snow removal is proposed by Liu et al. [44]. The snow removal network consists of a translucency and residual recovery module that handles the restoration of the snow-free image from semi-transparent and fully opaque snow streaks, respectively. The architecture is inherited from Inception-v4 [63] and enhanced by using the atrous spatial pyramid pooling from DeepLab [13].

3.2 Video-Based Rain Removal

The first attempts of removing rain in video sequences took advantage of the short duration of a rain streak, i.e. that a single streak is visible to the camera in one frame and then disappears. This means that the rain removal problem may be formalized as a low-pass filtering problem in which the rain streaks are unwanted high-frequency fluctuations. As such, the rain will be removed by applying a temporal median filter on the entire image [25,61]. The problem with this approach is that all other temporal motion will be blurred too.

Photometric Constraint

In order to prevent blurring of the non-rain image, it is therefore beneficial to detect the individual rain streaks. Such detection was introduced by K. Garg and S.K. Nayar in 2004 [22] when they studied the photometry of falling raindrops. As we described in Section 2, they introduced a method to find candidate rain pixels based on the observation that a raindrop appears brighter than the background and that each rain streak only appears in a single frame. Thus, under the assumption that the background remains stationary, the candidate pixels that may contain a rain streak should satisfy the following condition:

$$\Delta I = I_n - I_{n-1} = I_n + I_{n+1} \geq c \tag{F.10}$$

where I_n denotes the image at frame n, and c is the minimum intensity change to distinguish rain drops, fixed to $c = 3$. For each frame, the candidate streaks are refined by the requirement that the intensity change of pixels on the same

streak should be linearly related to the background intensities, I_b, at time $n-1$ and $n+1$:

$$\Delta I = -\beta I_b + \alpha \tag{F.11}$$

This should hold for all pixels within a single streak as imaged by a single frame if β is within the range $[0; 0.039]$ [22]. The step performed in Equation F.11 is denoted as the photometric constraint. However, in subsequent work on video-based rain removal, also the constraint of Equation F.10 is denoted as the photometric constraint. In our overview of rain removal methods in Table F.4, we use the term 'photometric constraint' if either Equation F.10 or F.11 have been applied. In the work by Garg and Nayar, the binary output of the photometric constraint is correlated for a temporal window of 30 frames. Spatio-temporal streaks that have a strong directional component are regarded as the detected rain streaks. In Table F.4, we refer to this, and variants hereof, as the streak orientation constraint. Streaks that consist of only a few pixels will be filtered out during this selection as their Binary Large OBjects (BLOBs) will not impose a strict directional structure.

The detected rain streaks are removed by using the two-frame average of frame $n-1$ and $n+1$, i.e. the temporal mean. Rain removal algorithms that use the photometric constraint, for example, are summarized in Table F.7. These algorithms typically include a separate detection and removal step, where detected rain pixels are smoothed out by using either a temporal or spatial filter. We denote this as the 'removal method' in Table F.7.

Chromatic Constraint

The intensity-based temporal constraint of Equation F.10 is usually applied on gray-scale images. In [80], the constraint is extended to color images by assuming that the temporal differences of the three color channels are approximately similar when the background is occluded by a rain streak, otherwise not.

Streak Orientation

A background subtraction algorithm is used in [6] to generate candidate streaks, which are refined by the selection rule of Equation F.10 and the removal of large BLOBs. The orientation of the remaining streak candidates is modeled by a Gaussian-uniform mixture distribution. By the assumption of the similar orientation of rain streaks, rain streaks are detected if the Gaussian part of the mixture distribution is dominant relative to the uniform part.

Barnum et al. [5] analyzed the properties of rain streaks in frequency space and found that the rain streaks impose a strong directional component in the Fourier-transformed image. By thresholding the rotation and magnitude of

the Fourier-transformed videos, they are capable of detecting most of the rain. As the rotation of rain streaks is dependent on the wind, one has to manually tune the ratios for each rainfall.

Matrix Decomposition

The intensity fluctuations with respect to a background model are used as an initial estimate of sparse rain streaks and the foreground in [52]. These are used as the initial estimates of a matrix decomposition problem, where the image is decomposed into background, foreground, dense streaks, and sparse streaks:

$$I = B + F + R_s + R_d \tag{F.12}$$

where I is the input image, B is the background, F is the foreground, and R_s, R_d are the sparse and dense rain streaks, respectively.

The decomposition is enabled by a Markov Random Field (MRF), which uses optical flow from adjacent frames to detect moving objects from which rain removal is performed by using similar patches in adjacent frames.

Jiang et al. [32] expanded the matrix decomposition problem into a tensor decomposition problem by integrating the adjacent frames in the decomposition. They assumed that rain streaks are vertical and thus proposed to minimize the l_0 and l_1 norm of the total variation on the x and y axes, respectively. It is, furthermore, assumed that the temporal difference between the rain-removed frames is minimal. These constraints are solved using the Alternating Direction Method of Multipliers (ADMM) [7].

Dictionary Learning

Instead of using the candidate pixel selection of Equation F.10, Kim et al. [36] used two-frame optical flow to generate the frame difference, which is used as the initial rain map. The rain map is decomposed into sparse basis vectors corresponding to patches of size 16×16 pixels. A pre-trained SVM classier is used to filter the rain streaks from noise based on the orientation of the patch. The rain-removed image is restored using rain-free patches from adjacent frames in an Expectation-Maximization (EM) scheme.

3.3 Rain Removal Benchmarks

As mentioned in the introduction, existing evaluations of rain removal algorithms are based on short video sequences or a collection of images from the authors. Quantitative evaluation is typically performed on a set of rain-free images, where synthetic rain is overlaid. The synthetic rain is either produced in Adobe Photoshop[3] [19], reused from the work of Garg and Nayar [23],

[3] http://www.photoshopessentials.com/photo-effects/rain/

Table F.7: Rain Removal Algorithms That do Not Include Decomposition, Dictionary Learning, or Neural Networks. I: Image, V: Video.

Paper	Year	Detection method				Removal method
		Input	Photometric constraint	Chromatic constraint	Streak orientation	
[54]	2015	V	X	X	X	Temporal blending
[51]	2014	I		X	X	Inpainting
[12]	2014	V	X	X		Spatio-temporal mean
[35]	2013	I			X	Nonlocal means filter
[67]	2013	V	X		X	Temporal mean
[81]	2013	I				Guided filter
[72]	2012	I				Guided filter
[74]	2012	V	X			Inpainting
[68]	2012	V	X			Temporal mean
[66]	2011	V	X		X	Temporal mean
[57]	2011	V	X			Spatio-temporal mean
[6]	2011	V	X		X	None
[82]	2011	V		X	X	Blending
[5]	2010	V			X	Temporal mean
[43]	2009	V	X	X		Kalman filter
[8]	2008	V	X		X	Temporal mean
[50]	2008	V	X		X	Temporal mean
[24]	2007	V	X		X	Temporal mean
[80]	2006	V		X	X	Temporal blended mean
[61]	2003	V				Temporal median
[25]	1999	V				Temporal median

which considers the photo-realistic rendering of single rain streaks, or produced by using own methods [5,65]. For videos, the synthetic rain is produced in Adobe After Effects [36].

The two most popular metrics for comparing the rain-removed image and the original rain-free image are the Structural Similarity Index (SSIM) [71] and the Peak Signal-to-Noise-Ratio (PSNR). Other common metrics include the Visual Information Fidelity (VIF) [56] and the Blind Image Quality Index (BIQI) [49]. BIQI is a no-reference algorithm for assessing the quality of an image, meaning that it does not require the rain-free image for comparison. Other metrics include the forward-backward feature tracking accuracy [5], the measurement of image variance [68], and the comparison of face detection scores on original and rain-removed images [54]. An overview of how the existing rain removal algorithms are evaluated is listed in Table F.8. It should be noted that we have only included comparisons with dedicated rain

Table F.8: Existing Evaluations of Rain Removal Algorithms. I: Image, V: Video.

Paper	Year		Comparison metrics				Compared methods
		Input	SSIM	PSNR	VIF	Other	
[79]	2017	I	X	X	X	X	[14,19,34,45,78]
[78]	2017	I	X	X			[14,19,34,45]
[19]	2017	I	X			X	[29,39,45]
[20]	2017	I	X				[39,45]
[32]	2017	V	X	X		X	[36,39,45]
[52]	2017	V		X			[24,36,68,80]
[75]	2017	I	X	X			[34,39,45]
[44]	2017	I	X	X			[19]
[10]	2017	I	X	X	X	X	[29,34]
[70]	2017	I	X	X		X	[11,39,45]
[39]	2016	I	X				[14,34]
[36]	2015	V		X			[5,24,34,80]
[45]	2015	I	X	X			[34,36]
[54]	2015	V				X	[24,80]
[11]	2014	I			X		[24,34]
[29]	2014	I					
[62]	2014	I	X	X			[29]
[51]	2014	I					[34]
[1]	2014	V					[24]
[12]	2014	V					[24,68,80]
[35]	2013	I					[34]
[67]	2013	V				X	[5,24,43,66,68]
[14]	2013	V			X		[5,34,72]
[81]	2013	I					[72]
[30]	2012	V		X			[34]
[34]	2012	I			X		
[72]	2012	I					
[74]	2012	V					[24]
[68]	2012	V				X	[24,43,66,80]
[21]	2011	I					
[66]	2011	V				X	[24,43,80]
[57]	2011	V					[24]
[6]	2011	V					
[82]	2011	V					
[5]	2010	V				X	[24,80]
[43]	2009	V					[24,80]
[8]	2008	V					[24]
[50]	2008	V					[24,80]
[24]	2007	V					
[80]	2006	V					[24]
[61]	2003	V					
[25]	1999	V					

removal algorithms and thus excluded comparisons with general-purpose noise removal or image filtering algorithms.

It is observed from Table F.8 that only a few competing rain removal al-

gorithms are evaluated for each proposed method, thus hindering a general comparison of the entire field. Fortunately, recent works on rain removal include a more thorough evaluation on competing algorithms. On average, the rain removal algorithms published in 2017 have been evaluated on approximately three competing algorithms. However, a true overview of the performance across algorithms remains a challenge. This is caused by the following:

1. Few authors have made their implementations publicly available.
2. There is limited availability of public datasets for validation.

The implementations of [20, 29, 30, 34, 36, 79] are available to the general public. If one wants to compare other methods, they must be re-implemented manually, which does not guarantee comparable performance nor comparable results.

A few public datasets have recently emerged. Li et al. [39] introduced a dataset with 12 images[4], all with and without artificial rain. Along with their open-source implementation of their proposed rain removal algorithm, Zhang et al. [79] also made their training and test sets available; these consist of a total of 800 images[5]. For video-based rain removal, unfortunately, no such dataset exists. Thus, any application-based evaluation of rain removal algorithms are hindered due to lack of appropriate datasets.

3.4 Common Challenges of Rain Removal Algorithms

In Table F.9, we summarize the underlying assumptions and the main challenges of the reviewed rain removal algorithms. It is interesting to note that even the sophisticated algorithmic methods of matrix decomposition and sparse dictionary are governed by heuristic assumptions that not necessarily translate to real-world conditions. The recent advent of CNNs in rain removal is promising but relies on a collection of synthetic images for training. The generation of more realistic, synthetic rain as well as the introduction of synthetic rain in longer video sequences could help move the frontiers in image and especially video-based rain removal.

4 New Dataset

In order to thoroughly evaluate the performance of rain removal algorithms under real-world conditions, we introduce the AAU RainSnow dataset[6] that includes 22 challenging sequences captured from traffic intersections in the

[4] http://yu-li.github.io/
[5] https://github.com/hezhangsprinter/ID-CGAN/
[6] `https://www.kaggle.com/aalborguniversity/aau-rainsnow`

Table F.9: Common Challenges of Rain Removal Algorithms.

Detection method	Image	Video	Assumptions	Problems
Spatial filtering	X		R high-frequency patterns	Textured objects detected as R
Temporal filtering		X	Static B, moving R	Moving objects detected as R
Photometric, chromatic constraint		X	Semi-static B, moving R	Moving light colored objects detected as R
Streak orientation	X	X	R fall in a particular direction	R orientation vary according to the wind
Matrix decomposition	X	X	R patches share unique descriptors	No guarantee that descriptors adhere to real world
Sparse dictionary	X		R patches share unique patterns	Some R patterns are shared with B
CNN	X	X	R patterns can be learned offline	Requires synthetic images for training

Danish cities of Aalborg and Viborg. The sequences of the dataset are captured from seven different locations with both a conventional RGB camera and a thermal camera, each with a resolution of 640 x 480 pixels at 20 frames per second. We provide color information for the conventional RGB camera as some rain removal algorithms are explicitly created for color images and some segmentation algorithms produce better results with color than gray-scale images [33].

Rain and snow are the common denominators of the sequences. In some sequences, the rain is very light and mostly visible as temporal noise. In other sequences, the rain streaks are clearly visible spatial objects. The illumination conditions vary from broad daylight to twilight and night. When combined with rain, snow, moist, and occasional puddles on the road, the variations in lighting create several challenging conditions, such as reflections, raindrops on the camera lens, and glare from headlights of oncoming cars at night.

The characteristics of each scene in our dataset are listed in Table F.10. The weather conditions and the temperature for each scene have been estimated by correlating the observed weather with publicly available weather station data[7]. The distance from the weather station to the scene is 25 km for the Ringvej sequences and a maximum of 13 km for all other sequences. We also include key characteristics of the BadWeather sequences in the ChangeDetection.net dataset, as shown in Table F.10, to enable a comparison of the two datasets.

One observes from Table F.10 that our dataset comprises of more objects per frame and that the observed objects are significantly smaller than the BadWeather sequences. Falling snow is present in all of the BadWeather sequences, but the lighting conditions are fine, with all areas of the scene being sufficiently lit. Thus, we believe that the detection and segmentation of objects pose a significant challenge in our proposed dataset. Image samples for every traffic intersection in our dataset are shown in Figure F.4.

4.1 Annotations

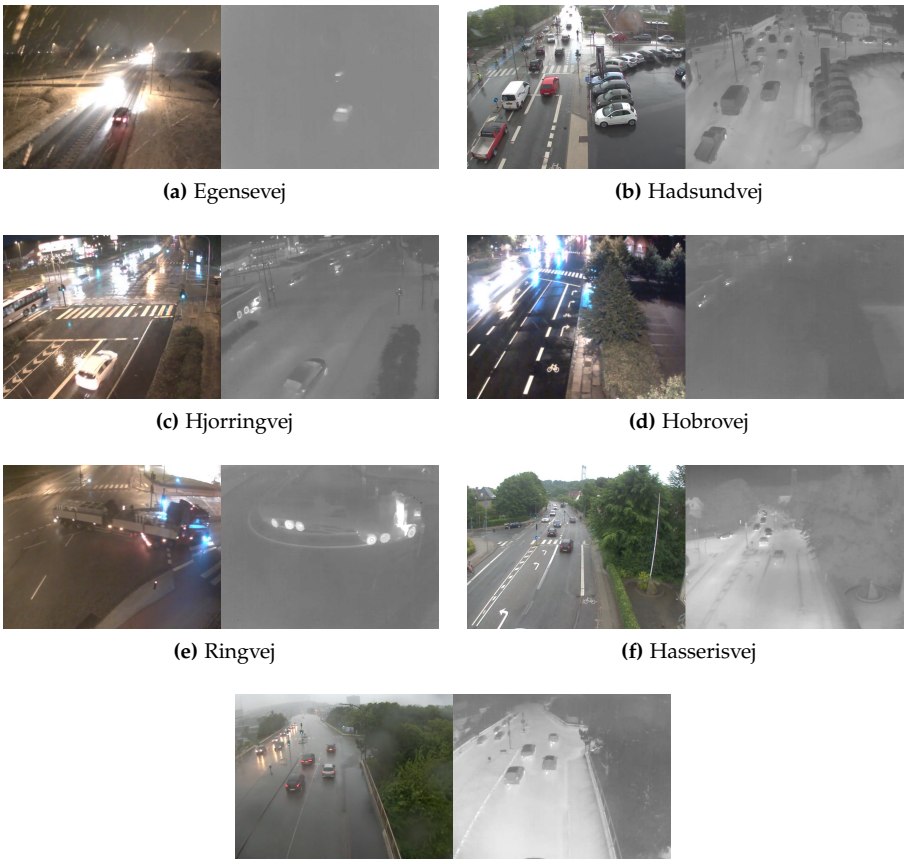
All frames of the BadWeather sequences are annotated at pixel level by the ChangeDetection.net initiative. For our dataset, the manual pixel-level annotation is complicated by the smaller size of apparent objects and many reflections. Thus, in order to make the annotations feasible, we have randomly selected 100 frames for each sequence from a uniform distribution. In a five-minute sequence at 20 frames per second, this means that, on average, an annotated frame is available every three seconds. We believe that the strong correlation between subsequent frames and the smooth motion of the road users enables us to achieve a good approximation of the entire sequence by annotating only a small subset of the frames. Consequently, we would rather

[7] https://www.wunderground.com/

Table F.10: Key Characteristics of the AAU RainSnow and The BadWeather (BW) ChangeDetection.net Datasets. The Approximate Duration of the BadWeather Sequences are Calculated with 20 Frames/Second. An 'L' in Rain or Thunderstorm indicates Light Rain and Light Thunderstorms, respectively.

Sequence	Time of day (24 H)	Duration (minutes)	Average number of objects per frame	Average object size (pixels)	Rain	Snow (X), (F)og	Thunderstorm	Estimated temp. (°C)
Egensevej-1	18:00	5.0	3.10	636	L			5
Egensevej-2	13:03	5.0	3.21	304	L	X		2
Egensevej-3	17:00	5.0	4.43	409	X			2
Egensevej-4	18:00	5.0	4.83	448		X		1
Egensevej-5	03:00	5.0	0.28	687	X			3
Hadsundvej-1	20:06	4.0	6.69	795			X	19
Hadsundvej-2	15:23	5.0	16.8	1293	X		L	13
Hjorringvej-1	06:04	5.0	3.48	1451	X			12
Hjorringvej-2	16:00	5.0	13.7	1589	L			12
Hjorringvej-3	19:12	5.0	6.62	1317	X			11
Hjorringvej-4	07:00	5.0	6.49	1926	L			8
Hobrovej-1	05:00	5.0	1.63	2438	L			14
Ringvej-1	20:05	5.0	3.5	1033	L			12
Ringvej-2	05:05	4.0	0.86	4533		F		5
Ringvej-3	17:54	5.0	3.88	983	L			14
Hasserisvej-1	13:12	5.0	4.85	560	X			11
Hasserisvej-2	10:00	5.0	5.75	703	L			19
Hasserisvej-3	13:00	5.0	7.58	616	L			19
Ostre-1	06:10	5.0	2.50	771	X			15
Ostre-2	10:00	5.0	10.7	896	L			10
Ostre-3	18:00	5.0	4.98	514	X		X	11
Ostre-4	19:20	5.0	3.10	672	X			10
BW/blizzard	-	5.8	0.73	2391		X		-
BW/skating	-	3.3	0.58	6522		X		
BW/snowFall	-	5.4	0.15	7803		X		-
BW/wetSnow	-	2.9	0.53	3261		X		-

spend time annotating more scenes that can capture a variety of challenging weather conditions than annotating a single sequence in its entirety. The annotation of our dataset is usually performed on the RGB images and mapped to the thermal images via a planar homography. In the case of severe reflections, the thermal image is used to guide the annnotations instead. We use the AAU VAP Multimodal Pixel Annotator [2] for drawing the annotations. We



(a) Egensevej **(b)** Hadsundvej

(c) Hjorringvej **(d)** Hobrovej

(e) Ringvej **(f)** Hasserisvej

(g) Ostre

Fig. F.4: Samples of each of the seven traffic intersections of the AAU RainSnow dataset. One sample is shown for each sequence with corresponding RGB and thermal image. For improved visibility, contrast is adjusted for all thermal images, except for the Hadsundvej sequence.

have marked 'do not care' care zones in areas without road users and when all objects are very small in a particular region, for instance the top of the surveillance video. Examples hereof are shown in Figure F.13b.

5 Evaluation Protocol

We will evaluate whether the rain removal algorithms introduced in Section 3 make a difference when used in a traditional computer vision pipeline that includes segmentation, tracking, and instance segmentation. In other words, a successful rain removal algorithm should improve the ability of subsequent

algorithms to segment objects and perform feature tracking. In the context of traffic surveillance, the objects are road users. As such, the visual quality of the rain removed images or videos is not a concern as long as the subsequent traffic surveillance algorithms improves.

If one wants to assess the visual quality of the rain removed images, the mean opinion score from multiple human assessments could be used. A user study is conducted in [70] to consider the most favorable result of several rain removal algorithms.

We use the AAU RainSnow dataset and the BadWeather sequences described in Section 4 as the evaluation dataset. In order to run the rain removal algorithms on the datasets, the implementation should be available. However, for most of the algorithms listed in Table F.4, the implementation is not publicly available. Additionally, in most cases, it is not possible to re-implement the algorithms due to missing details in the original papers. Fortunately, the implementations of:

- Zhang et al. [79]
- Fu et al. [20]
- Kim et al. [36]
- Kang et al. [34]

are publicly available and will be used for comparison. Furthermore, we have implemented the rain removal algorithm by Garg and Nayar [24] as their work is generally considered the cornerstone from which many video-based rain removal algorithms are built. Our implementation is publicly available[8] and also provides links to the implementations listed above.

As the baseline for rain removal, we have added a spatial 3×3 pixels mean filter, which makes the image more smooth and may reduce the amount of rain. The evaluated rain removal algorithms are listed in Table F.11. We use the default parameters from the original papers and list the average image processing time for every algorithm.

It should be noted that, although the dataset consists of video sequences, we have included both single-image and video-based rain removal algorithms. Although it is the general impression that video-based rain removal is significantly easier than single-image based rain removal, it has not been experimentally verified whether video-based algorithms use this advantage to outperform single-image based methods. Thus, we would like to find out by including algorithms from both categories.

The rain removal algorithms of Table F.11 undergo a two-tied evaluation on a segmentation, instance segmentation, and feature tracking pipeline. In the following, we will describe the protocol of the three evaluation pipelines.

[8] https://bitbucket.org/aauvap/rainremoval

Table F.11: Evaluated Rain Removal Algorithms. The Processing Time Per Image is Measured on an Intel Core i7-3770 CPU with NVIDIA 1080Ti Graphics. *The Method from Zhang2017 is GPU-bound, all Other Methods are CPU-bound.

Paper	AuthorYear	Input	Main detection method	Learning	Processing time per image
[79]	Zhang2017	Image	Conditional Generative Adversarial Network (DCGAN)	Offline	0.40 s*
[20]	Fu2017	Image	Convolutional Neural Network	Offline	0.69 s
[36]	Kim2015	Video	Sparse dictionary	Offline	31.12 s
[34]	Kang2012	Image	Sparse dictionary	Online	78.72 s
[24]	Garg2007	Video	Photometric constraint, streak orientation, size	Fixed	0.81 s
-	'Median'	Image	Spatial 3×3 pixels mean filter	Fixes	0.06 s

5.1 Segmentation

We evaluate the performance of rain removal algorithms under a traditional segmentation pipeline by running the rain removal algorithms in a separate pass and then running the segmentation algorithms on top of the rain-removed imagery. In order to select a segmentation algorithm that is representative of the state-of-the-art, we look to the results of the ChangeDetection.net challenge [69]. Although recent advantages in convolutional neural networks have led to superior performance of supervised segmentation methods as seen in Table F.3, we turn to the unsupervised methods instead. We believe that, in order for a segmentation method to be applicable in a real-world traffic surveillance context, the method should work out-of-the-box for non-experts and not require hand tuning in the form of parameters or training samples.

A representative of a top 3 unsupervised segmentation method is the 'SuBSENSE' algorithm [60]. SuBSENSE is a method that builds on the spatial diffusion step introduced in ViBE [4]. Instead of relying only on color information for the pixel description, SuBSENSE includes information of the local neighbors by computing Local Binary Similarity Patterns (LBSP) for every pixel. Based on a majority vote of the LBSP and local pixel values, the pixel is classified as either foreground or background.

Furthermore, we include the Mixture of Gaussians (MoG) method as modified by Zivkovic [83] as this is a classic segmentation method that is well understood and often used as a baseline for comparisons.

We use the F-measure to measure and compare the performance of the rain

removal algorithms in the segmentation context. The F-measure is a widely used metric for evaluating change detection algorithms and has been found to agree well with the overall ranking computed from several metrics [69]. The F-measure is computed as:

$$F = 2 \cdot \frac{\mathrm{Pr} \cdot \mathrm{Re}}{\mathrm{Pr} + \mathrm{Re}} \tag{F.13}$$

where Pr (precision) is defined as:

$$\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{F.14}$$

and Re (recall) is defined as:

$$\mathrm{Re} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{F.15}$$

where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negative classified pixels in a sequence. The balance between recall and precision might be fine-tuned by adjusting the intrinsic parameters of the segmentation methods. For these experiments, we use the settings of the original authors. The evaluation of segmentation is performed as follows:

for Every video sequence in Table F.10 **do**
  Run the segmentation algorithms on the unmodified, original frames of the video
  Compute the F-measure of the segmented frame and the ground truth
  for Every rain removal algorithm in Table F.11 **do**
    Run the rain removal algorithm on each frame and save the rain-removed frame
    Run the segmentation algorithms on the rain-removed frames and save the result
    Compute the F-measure of the segmented frame and the ground truth
  end for
end for

The results hereof are presented in Section 6.1.

5.2 Instance Segmentation

The traditional segmentation challenge handles the separation of foreground objects from the background. In instance segmentation, one also needs to differentiate between every single instance of an object and assign the correct class label.

For evaluation, we follow the philosophy that algorithms should run out-of-the-box with no fine-tuning on our dataset. We use the popular Microsoft

COCO API [40] and report results in the main COCO competition metric, average precision (AP) measured over intersection over union ratios in the range from 0.50 to 0.95 with intervals of 0.05. We have selected two instance segmentation algorithms for evaluation: Fully Convolutional Instance Segmentation (FCIS) [38] which won the 2016 COCO segmentation challenge and Mask R-CNN [26] which outperformed the FCIS network and ranked 3rd in the 2017 COCO segmentation challenge. Both algorithms are trained on the ImageNet [15] and COCO [40] datasets. Experimental results showed that the assigned class labels from both instance segmentation algorithms did not agree well with the ground truth of our RainSnow dataset. For instance, most cars were classified as trucks. Therefore, we decided to measure the precision of the class-agnostic instance segmentation by setting the 'useCats' parameter of the COCO API to 'false'. The evaluation is only performed on the AAU RainSnow dataset, as the ChangeDetection.net dataset is incompatible with the COCO evaluation format.

5.3 Feature Tracking

We adopt the forward-backward feature-point tracking method used by Barnum et al. [5]. For every n frames in a sequence, we select the 200 strongest features [59] and track them in the next m frames using the Lucas-Kanade tracker [3]. After m frames, the features are tracked when the sequence is played backwards to the point in time where the features were instantiated. We calculate the tracking accuracy by measuring the distance between the start and end positions of the tracked features. Similar to [5], we report the number of successfully tracked features within an error margin of 1 and 5 pixels.

Inspired by [5], we have chosen the forward-backward tracking point accuracy for the following reasons:

- The tracking accuracy is correlated to the ability of the rain removal algorithms to preserve non-rain high-frequency components.
- The measure does not require ground truth and thus scales with the length of the sequence.
- If the tracking of a feature point is confused by the spatio-temporal fluctuations of a rain streak, the tracking accuracy should improve on rain-removed imagery.

Barnum et al. evaluate the tracking accuracy once on the entire sequence, i.e. $n = l$ and $m = l$, where l is the length of the video sequence, approximately five seconds [5]. As our sequences are much longer, we need many separate instances of the forward-backward feature tracking. We have empirically found $n = 1.5$ s and $m = 12$ s, meaning that for every 1.5 seconds, we select

the 200 strongest features which are tracked for 12 seconds forwards, then backwards. In our experience, different values of n and m only change the magnitude of the results.

The results of the feature tracking are presented in Section 6.3.

6 Results

As described in Section 5, we evaluate the rain removal algorithms of Table F.11 with respect to the performance of segmentation, instance segmentation, and feature tracking algorithms on the rain-removed sequences of Table F.10.

6.1 Segmentation

The segmentation results, as indicated by the F-measure for the Mixture of Gaussians (MoG) and the SuBSENSE (SuB) segmentation algorithms, are listed in Table F.12 and plotted in Figure F.5b. If we take a look at the segmentation results on the unmodified video, i.e. the non-rain-removed frames, we may note that he proposed AAU RainSnow dataset imposes a significant challenge to segmentation algorithms. The F-measure of our dataset varies from 0.11 to 0.66, even for the state-of-the-art SubSENSE method. The MoG method fare even worse, with F-measures in the range from 0.13 to 0.34. Segmentation results are much better on the BadWeather sequences, where the F-measure is in the range of 0.80 to 0.89 for the SuBSENSE method.

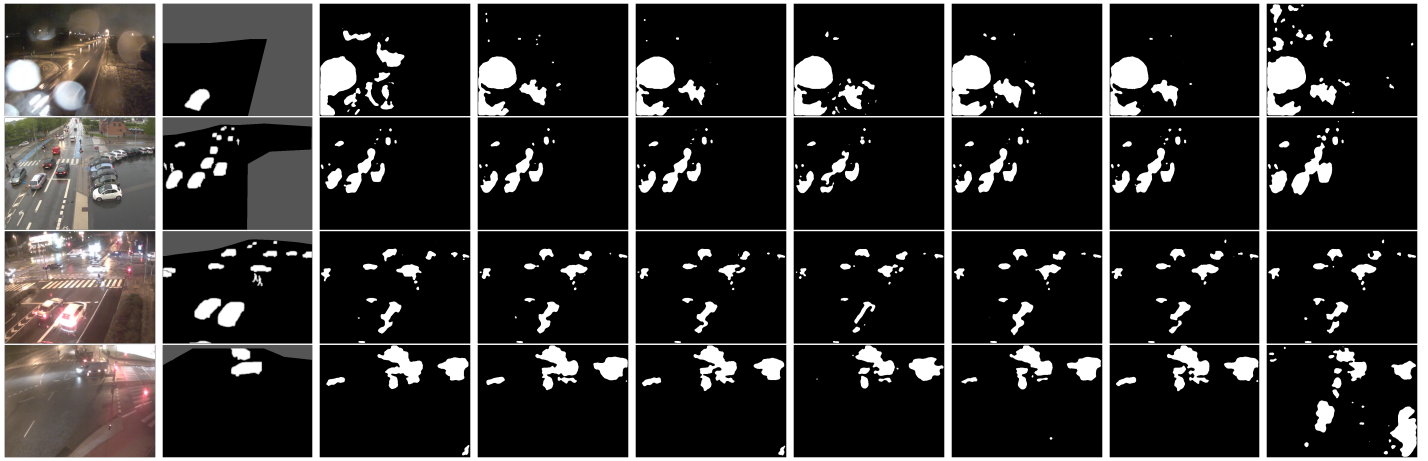
When looking at the segmentation results of the rain-removed images, we should take note of the aforementioned differences in segmentation performance and the inherent differences between the AAU RainSnow and BadWeather datasets as described in Section 4. On the AAU RainSnow dataset, we see from Table F.12 that the GAN-based convolutional neural network by Zhang et al. [79] gives an average increase of 28.5% in the segmentation performance of the SuBSENSE algorithm, whereas the same algorithm results in an average decrease of 38.6% on the BadWeather sequences. Except for the combination of MoG on the rain-removed videos in the method by Kim et al. [36], all rain removal algorithms reduce the performance of segmentation algorithms on the BadWeather dataset. Nevertheless, all rain removal algorithms give a performance increase when using the SuBSSENSE method based on the AAU RainSnow dataset. Examples of the visual segmentation results on the AAU Rainsnow database are shown in Figure F.13.

It is difficult to give an unequivocal explanation of the cause of the great difference seen in results between the two datasets. This variance may be caused by a combination several factors:

- The segmentation of the BadWeather sequences already produced good results, rendering it difficult to improve.

Table F.12: Evaluation of Segmentation Performance on Each Sequence.
absolute F-measure is reported for the original, non-rain-removed frames. Other results are relative to the original results of each sequence, in percentages. MoG: Mixture of Gaussians [83]. SuB: SuBSENSE [60]. Best result of a sequence is indicated in bold. Category averages are computed from the sum of absolute F-measures.

Sequence	Original		Median		Garg2007 [24]		Kang2012 [34]		Kim2015 [36]		Fu2017 [20]		Zhang2017 [79]	
	MoG	SuB	MoG	SuB	MoG	SuB	MoG	SuB	MoG	SuB	MoG	SuB	MoG	SuB
Egensevej-1-5	0.13	0.11	-5.77	13.9	-1.50	26.2	-21.4	12.0	**8.69**	17.4	-0.29	18.4	-4.20	**37.9**
Hadsundvej-1-2	0.19	0.66	-10.8	4.48	-7.67	2.50	-27.1	-9.46	1.41	3.78	-4.15	6.92	**14.8**	**21.4**
Hasserisvej-1-3	0.34	0.61	-4.16	10.6	-1.41	12.7	-4.38	11.4	-4.17	12.1	0.44	18.6	**10.7**	**24.4**
Hjorringvej-1-4	0.21	0.52	-1.57	7.73	-1.57	6.19	-20.2	5.48	-1.58	6.64	1.86	9.54	**14.7**	**22.1**
Hobrovej-1	0.28	0.36	1.61	-1.93	4.52	9.02	-36.8	**42.7**	8.75	21.8	**10.8**	0.63	3.93	-23.8
Ostre-1-4	0.26	0.49	10.5	10.9	10.8	13.2	-27.4	-9.87	11.3	12.3	14.6	14.8	**28.4**	**23.2**
Ringvej-1-3	0.23	0.19	16.1	5.40	14.6	13.9	-0.39	**76.0**	10.7	25.0	16.9	8.89	**23.7**	5.51
BadWeather/blizzard	0.25	0.85	-98.6	-99.7	-99.0	-99.7	-90.5	-91.1	-5.03	-1.43	-7.17	-1.47	**15.5**	**0.53**
BadWeather/skating	0.28	0.89	-3.57	-7.39	7.33	-11.9	-76.7	-87.2	**9.22**	**-7.30**	-6.84	-7.75	-79.3	-91.4
BadWeather/snowFall	0.18	0.88	0.57	-18.3	5.87	-18.1	-74.8	-90.1	32.4	**-15.8**	12.2	-17.3	**38.2**	-18.9
BadWeather/wetSnow	0.25	0.80	-0.23	-20.3	2.96	-26.3	-87.7	-95.3	**16.7**	-19.9	5.11	**-18.5**	-37.4	-47.9
AAU RainSnow avg.	0.22	0.40	-0.16	6.36	1.31	9.28	-18.6	10.5	4.55	3.42	4.29	10.5	**14.0**	**28.5**
BadWeather avg.	0.25	0.87	-31.5	-37.0	-24.4	-39.4	-82.4	-89.5	**9.88**	**-8.75**	-2.33	-9.15	-18.7	-38.6



(a) RGB **(b)** Ground Truth **(c)** Original **(d)** Median **(e)** Garg2007 **(f)** Kang2012 **(g)** Kim2015 **(h)** Fu2017 **(i)** Zhang2017

Table F.13: Segmentation results on the AAU RainSnow dataset by the SuBSENSE algorithm [60]. Each row represents the results of different rain removal algorithms on a single frame. Sequences from top to bottom: Egensevej-5, Hadsundvej-1, Hjorringvej-4, and Ringvej-2. Gray areas indicate don't care zones.

- The average object size of the BadWeather sequences is 3.5 times larger than the average object size of the AAU RainSnow dataset. This difference may occur because the segmentation of smaller objects benefits from the removal of rain streaks, whereas the segmentation of larger objects is more resilient to the fluctuations from rain and snow. In this case, the spatio-temporal low-pass filtering of the rain removal algorithms may eventually harm the segmentation performance.

It should be noted that further experimentation on other datasets is needed in order to fully understand the underlying causes.

The visual results of Figure F.13 based on the AAU RainSnow dataset confirm that raindrops on the lens and reflections on the road pose a challenge to the segmentation process. However, even under these challenging conditions, the results from Table F.12 show that the evaluated rain removal algorithms improve the segmentation. Two notable exceptions are the Egensevej-5 and Ringvej-2 sequences, which are shown in the top and bottom rows of Figure F.13, respectively. On the Egensevej-5 sequence, the best rain removal algorithm decreases the segmentation performance of the SuBSENSE algorithm by 44%, whereas the remaining algorithms perform even worse. On the Ringvej-2 sequence, the otherwise top performing rain removal methods of Fu [20] and Zhang [79] fail to improve the segmentation results at all. One possible explanation could be that the segmentation performance of the two original sequences is quite poor, with F-measures of 0.05 and 0.14 for the SuBSENSE method on the Egensevej-5 and Ringvej-2 sequences, respectively. If the underlying phenomena responsible for the degradation of the visual quality are not related to rainfall and snowfall, the corrections from rain removal algorithms may be ill-behaved and degrade the results.

However, the remaining sequences of the AAU RainSnow dataset show decent increases in segmentation performance for most rain removal algorithms, even for sequences in which the segmentation is relatively hard. The improvement in segmentation results from relatively 'good' sequences with good illumination and few shadows, such as the Hadsundvej sequences, indicates that rain removal algorithms could be a suitable preprocessing step for traffic surveillance scenes under rain and snow when improved performance of subsequent traditional segmentation algorithms is required.

6.2 Instance Segmentation

The average precision of the instance segmentation methods Mask R-CNN [26] and FCIS [38] are shown in Table F.14 and visualized with box plots in Figure F.5c. It is evident from the results of the original sequences that the Mask R-CNN method outperforms the FCIS method by a large margin, resulting in a AP of 0.33 and 0.07 on the entire AAU RainSnow dataset, respectively. If

we compare the instance segmentation results with the traditional segmentation results of Table F.12, both segmentation approaches struggle with the Egensevej sequences. On the Hobrovej sequence, the traditional segmentation methods fares well whereas the instance segmentation methods breaks down. One should note, however, that the instance segmentation methods does not take temporal information into account, which makes the segmentation increasingly harder under difficult weather.

All evaluated rain removal algorithms fail to improve the instance segmentation results of the Mask R-CNN. The best performing algorithm of Zhang et al. [79] degrades the AP by 3.45% while the worst performing method by Kang et al. [34] degrades the result by 36.3%.

On the contrary, all rain removal algorithms but the method by Kang et al. improves the instance segmentation results of the FCIS method. The best rain removal algorithm on the FCIS method is the 3x3 spatial mean filter and the CNN-based method by Zhang et al. which improves the result by 27.6% and 25.2%, respectively. However, even with these improvements, the FCIS method is inferior to Mask R-CNN.

It is remarkable that the simple median filter outperforms the dedicated rain removal algorithms with the FCIS method and lies close to other algorithms with the Mask R-CNN. Both instance segmentation methods have been trained on the ImageNet and COCO datasets and are thus designed to respond to images that resemble these training sets. Our AAU RainSnow dataset is a different, surveillance-type dataset with many small objects that does not necessarily resemble these training datasets. Given a dissimilar dataset, the noise and alterations by the applied rain removal algorithms might push the images out of the visual manifold that the instance segmentation methods have been trained on.

6.3 Feature Tracking

The results of the forward-backward feature point tracking are shown in Table F.15 and the box plots of Figure F.5d. If we look at the average results on both datasets, it is observed that the rain removal algorithm by Zhang et al. [79], which was superior when evaluated on the segmentation pipeline, consistently deteriorates the feature tracking performance. It should be noted that the algorithm by Zhang et al. is a single-frame based method and does not incorporate the temporal information when removing the rain. In fact, all the evaluated single-frame rain removal algorithms deteriorate the feature-tracking results (Median, Kang et al. [34], Fu et al. [20], Zhang et al. [79]).

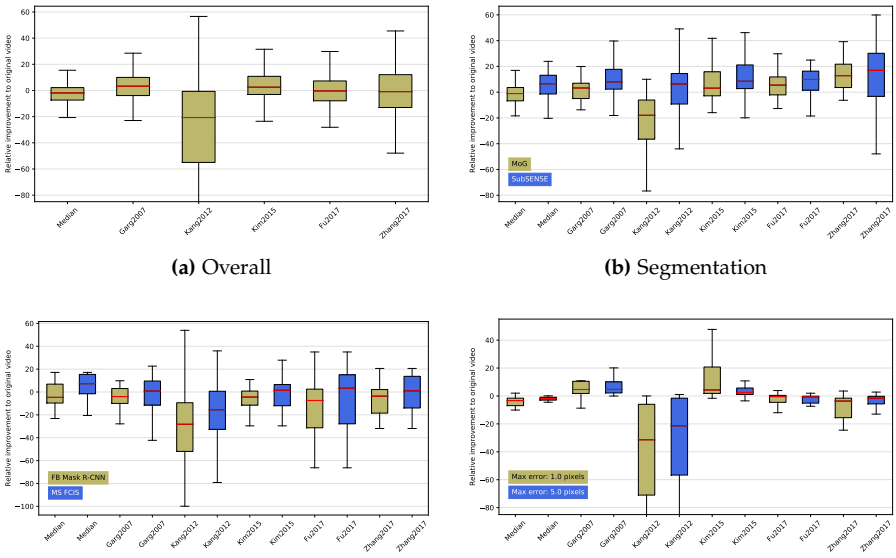
If we look at the results of the video-based rain removal methods by Garg and Nayar [24] and Kim et al. [36], we observe a general increase in feature tracking performance. The relatively simple method by Garg and Nayar contributes to an average increase in the number of successfully tracked

Table F.14: Evaluation of Instance Segmentation Performance on Each Sequence.
average precision (AP[.5:.05:.95] is reported for the original, non-rain-removed frames. Other results are relative to the original results of each sequence, in percentages. FB: Mask R-CNN from Facebook [26]. MS: FCIS from Microsoft [38]. Best result of a sequence is indicated in bold.

Sequence	Original		Median		Garg2007 [24]		Kang2012 [34]		Kim2015 [36]		Fu2017 [20]		Zhang2017 [79]	
	FB	MS	FB	MS	FB	MS	FB	MS	FB	MS	FB	MS	FB	MS
Egensevej-1-5	0.10	0.02	-9.32	2.30	**-7.27**	-23.3	-49.3	-8.33	-10.1	-6.97	-28.4	-55.8	-23.2	**4.86**
Hadsundvej-1-2	0.45	0.07	-4.99	**95.3**	-3.93	86.8	-26.6	50.4	-3.41	92.7	-29.0	58.9	**1.76**	89.3
Hasserisvej-1-3	0.44	0.12	-6.91	1.48	-5.76	-0.95	-9.70	0.31	-5.46	0.71	-0.04	4.18	**0.88**	**5.43**
Hjorringvej-4-1	0.34	0.10	-6.28	-0.38	-7.27	-4.03	-38.3	-15.7	-6.76	**-0.04**	-44.2	-26.1	**-6.03**	-1.79
Hobrovej-1	0.00	0.00	-49.9	17.1	**219.1**	**342.5**	-79.6	-15.2	-60.9	65.8	-82.1	-31.9	-27.5	-87.4
Ostre-1-4	0.29	0.07	-14.7	7.51	-4.41	4.20	-68.2	-60.4	-4.67	3.64	-6.51	6.47	**-2.19**	**15.5**
Ringvej-1-3	0.27	0.05	-8.52	3.47	-12.6	-13.5	-53.5	-56.5	-10.5	-5.24	**-3.63**	**17.5**	-25.5	-27.4
AAU RainSnow avg.	0.33	0.07	-7.70	**27.6**	-5.89	21.7	-36.3	-1.83	-5.77	25.2	-24.5	11.4	**-3.45**	24.9

Table F.15: Evaluation of Forward-Backward Feature Tracking on Each Sequence.
number of successfully tracked features with an error margin of 1.0 and 5.0 pixels is reported for the original, non-rain-removed frames. Other results are relative to the original results of each sequence, in percentages. Best result of a sequence is indicated in bold. Category averages are computed from the sum of tracked features.

Sequence	Original		Median		Garg2007 [24]		Kang2012 [34]		Kim2015 [36]		Fu2017 [20]		Zhang2017 [79]	
	1.0	5.0	1.0	5.0	1.0	5.0	1.0	5.0	1.0	5.0	1.0	5.0	1.0	5.0
Egensevej-1-4	44601	64486	-5.51	-3.31	21.6	22.9	-68.2	-57.8	**52.5**	**32.1**	-3.20	-5.76	-27.6	-21.6
Hadsundvej-1-2	46558	54592	-3.17	-1.84	**5.07**	**4.63**	-7.95	-3.61	3.39	2.25	0.12	-0.01	-1.33	-0.40
Hasserisvej-1-3	88155	94372	-2.52	-0.81	**2.19**	**2.73**	-0.40	0.08	1.52	1.88	-1.53	-0.85	-3.08	-1.84
Hjorringvej-1-4	87324	104680	-2.88	-1.90	**5.74**	**7.52**	-7.42	-2.21	5.20	4.68	0.19	-0.35	0.25	0.86
Hobrovej-1	32364	35502	-1.53	-0.62	**5.00**	**1.59**	-19.6	-9.67	4.07	1.03	0.26	-0.11	-4.53	-0.35
Ostre-1-4	106934	129240	-2.91	-1.77	**5.21**	**3.21**	-55.1	-33.5	2.72	0.27	-7.37	-6.98	-2.48	-0.81
Ringvej-1-3	67809	75810	-1.98	-1.05	1.44	1.42	-32.8	-19.7	**2.23**	**1.53**	-3.15	-2.90	-6.54	-1.93
BadWeather/blizzard	7630	17996	-77.1	-48.7	-8.73	**15.0**	-85.7	-60.8	**111.2**	5.95	-25.5	-18.4	-58.7	-25.7
BadWeather/skating	1934	12002	-39.9	-12.8	315.7	**18.1**	-100.0	-100.0	**455.9**	15.7	61.0	1.97	-45.6	-17.4
BadWeather/snowFall	1649	9983	-20.7	-15.8	276.6	98.0	-100.0	-100.0	**1093.8**	**163.8**	-12.0	-3.90	-36.9	-6.29
BadWeather/wetSnow	9571	15483	2.02	-1.41	45.9	**4.26**	-2.48	-3.97	**47.7**	2.72	9.98	-1.79	-3.18	-0.84
AAU RainSnow avg.	473745	558682	-2.87	-1.65	5.72	**6.01**	-27.1	-18.5	**7.72**	5.45	-2.63	-2.89	-5.07	-3.15
BadWeather avg.	20784	55464	-32.7	-21.8	69.3	27.6	-49.8	-60.5	**192.0**	**35.6**	-0.06	-6.75	-30.2	-13.5



(a) Overall **(b)** Segmentation

(c) Instance Segmentation **(d)** Feature Tracking

Fig. F.5: Box and whiskers plot of the relative improvement as a result of the pre-processing by rain removal algorithms.

feature points with a margin of error of 1 pixel of 5.72% and 69.3% on the AAU RainSnow and BadWeather datasets, respectively. Comparatively, the rain removal algorithm by Kim et al. [36] results in a modest improvement of 7.72% on the AAU RainSnow dataset. On the BadWeather dataset, the improvement is more pronounced with a corresponding performance increase of 192%. If we look at the average processing times per image listed in Table F.11, the method by Garg and Nayar is 38 times faster than the method by Kim et al. and should thus be preferred due to superior speed.

As opposed to the segmentation results, which did not agree on the AAU RainSnow and BadWeather datasets, the feature tracking results on the AAU RainSnow and BadWeather datasets differ only by an order of magnitude.

In general, the results indicate that feature-point tracking on traffic surveillance videos benefits from the spatial low-pass filtering of the video-based rain removal algorithms.

7 Conclusion

We have studied the effects of rain and snow in the context of traffic surveillance and reviewed single-frame and video-based algorithms that artificially remove rain and snow from images and video sequences. The study shows

that most of these algorithms are evaluated on synthetic rain and short sequences with real rain and their behaviour in a realistic traffic surveillance context are undefined and not experimentally validated. In order to investigate how they behave in the aforementioned context, we have presented the AAU RainSnow dataset that features traffic surveillance scenes captured under rainfall or snowfall and challenging illumination conditions. We have provided annotated ground truth for randomly selected image frames of these sequences in order to evaluate how the preprocessing of the input video by rain removal algorithms will affect the performance of subsequent segmentation, instance segmentation, and feature tracking algorithms.

Based on their dominance in the field and their public availability, we selected six rain removal algorithms for evaluation, two video-based methods and four single-frame based methods. The results presented in Table F.12 show that the single-frame based rain removal method of Zhang et al. [79] improves the segmentation by 19.7% on average on the AAU RainSnow dataset. However, it deteriorates the performance on the BadWeather sequences of the public ChangeDetection.net dataset [69] and is not successful on a classical feature tracking pipeline. As a result, we achieve lower accuracy on forward-backward feature tracking on the rain-removed frames by Zhang et al. than running the feature tracking on the unmodified original input frames. On the contrary, all video-based rain removal algorithms consistently improve the feature tracking results on the AAU RainSnow and BadWeather datasets. We received mixed results from the evaluation of instance segmentation methods. On a state-of-the-art method, the pre-processing by the evaluated rain removal algorithms decreased the segmentation performance. However, with the exception of the rain removal algorithm of Kang et. al. [34], all rain removal algorithms improved the performance on a slightly older, less capable instance segmentation method.

If we look at the overall improvement across the three evaluation metrics as shown in Figure F.5a, we observe a large variability in the performance of the rain removal algorithms. The simplest method, the spatial median filter, shows the lowest variability whereas the method from Kang et al. [34] shows the greatest variability and worst performance with a median improvement of -20%. The CNN-based methods of Fu et al. [20] and Zhang et al. [79] both show a median improvement around 0%, with lower variability of the former method. The video-based methods of Garg and Nayar [24] and Kim et al. [36] show similar performance with a median improvement at 3.3 and 2.5%, respectively. When considering the processing time required by the method of Kim et al., the well-established method from Garg and Nayar is considered to be the best general-purpose rain removal algorithm.

In this paper, we aimed to answer the initial research question: Does rain removal in traffic surveillance matter? We must conclude that, as with other aspects of computer vision, this really depends on the application.

Our experiments show that some applications benefit from rain removal, whereas other applications see their performance significantly reduced. Thus, rain removal algorithms should not be used as a general pre-processing tool in traffic surveillance, but they could be considered depending on the experimental results of the desired application. It should be noted that we have only tested the rain removal algorithms on video sequences in which it was actually raining or snowing. The behaviour of these algorithms on non-rain sequences is still undefined. Further investigations could go into an intelligent switching system that enables such pre-processing systems based on the available contextual information.

In our experiments, we have chosen to evaluate the performance on three computer vision methods: segmentation, instance segmentation, and feature tracking. However, it is still an open question how rain removal algorithms perform when evaluated on other methods, such as classification, object tracking, and 3D reconstruction.

Acknowledgment

The project has received funding from the Horizon 2020, the European Union Framework Programme for Research and Innovation (grant no. 635895). This publication reflects only the author's views. The European Commission is not responsible for any use that may be made of the information it contains.

References

[1] A. E. Abdel-Hakim, "A novel approach for rain removal from videos using low-rank recovery," in *Intelligent Systems, Modelling and Simulations, 5th International Conference on*. IEEE, 2014, pp. 351–356.

[2] C. H. Bahnsen, A. Møgelmose, and T. B. Moeslund, "The aau multimodal annotation toolboxes: Annotating objects in images and videos," *arXiv preprint arXiv:1809.03171*, 2018.

[3] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[4] O. Barnich and M. Van Droogenbroeck, "Vibe: A universal background subtraction algorithm for video sequences," *Image Processing, IEEE Transactions on*, vol. 20, no. 6, pp. 1709–1724, 2011.

[5] P. C. Barnum, S. Narasimhan, and T. Kanade, "Analysis of rain and snow in frequency space," *International Journal of Computer Vision*, vol. 86, no. 2, pp. 256–274, 2010.

[6] J. Bossu, N. Hautière, and J.-P. Tarel, "Rain or snow detection in image sequences through use of a histogram of orientation of streaks," *International Journal of Computer Vision*, vol. 93, no. 3, pp. 348–367, 2011.

[7] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[8] N. Brewer and N. Liu, "Using the shape characteristics of rain to identify and remove rain from video," in *Statistical Techniques in Pattern Recognition and Structural and Syntactic Pattern Recognition, Joint IAPR International Workshops on*. Springer, 2008, pp. 451–458.

[9] N. Buch, S. Velastin, J. Orwell *et al.*, "A review of computer vision techniques for the analysis of urban traffic," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 3, pp. 920–939, 2011.

[10] B.-H. Chen, S.-C. Huang, and S.-Y. Kuo, "Error-optimized sparse representation for single image rain removal," *Industrial Electronics, IEEE Transactions on*, vol. 64, no. 8, pp. 6573–6581, 2017.

[11] D.-Y. Chen, C.-C. Chen, and L.-W. Kang, "Visual depth guided color image rain streaks removal using sparse coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 8, pp. 1430–1455, 2014.

[12] J. Chen and L.-P. Chau, "A rain pixel recovery algorithm for videos with highly dynamic scenes," *Image Processing, IEEE Transactions on*, vol. 23, no. 3, pp. 1097–1104, 2014.

[13] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *arXiv preprint arXiv:1606.00915*, 2016.

[14] Y.-L. Chen and C.-T. Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," in *IEEE International Conference on Computer Vision, Proceedings of the*, 2013, pp. 1968–1975.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, IEEE Conference on*. IEEE, 2009, pp. 248–255.

[16] D. Eigen, D. Krishnan, and R. Fergus, "Restoring an image taken through a window covered with dirt or rain," in *International Conference on Computer Vision, Proceedings of the IEEE*, 2013, pp. 633–640.

[17] M. J. Fadili, J.-L. Starck, J. Bobin, and Y. Moudden, "Image decomposition and separation using sparse representations: an overview," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 983–994, 2010.

[18] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

[19] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *Image Processing, IEEE Transactions on*, vol. 26, no. 6, pp. 2944–2956, 2017.

[20] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *Computer Vision and Pattern Recognition, IEEE Conference on*, July 2017.

[21] Y.-H. Fu, L.-W. Kang, C.-W. Lin, and C.-T. Hsu, "Single-frame-based rain removal via image decomposition," in *Acoustics, Speech and Signal Processing, IEEE International Conference on*. IEEE, 2011, pp. 1453–1456.

[22] K. Garg and S. K. Nayar, "Detection and removal of rain from videos," in *Computer Vision and Pattern Recognition, 2004 IEEE Conference on*, vol. 1. IEEE, 2004, pp. I–I.

[23] ——, "Photorealistic rendering of rain streaks," in *Graphics, ACM Transactions on*, vol. 25, no. 3. ACM, 2006, pp. 996–1002.

[24] ——, "Vision and rain," *International Journal of Computer Vision*, vol. 75, no. 1, p. 3, 2007.

[25] H. Hase, K. Miyake, and M. Yoneda, "Real-time snowfall noise elimination," in *Image Processing, International Conference on*, vol. 2. IEEE, 1999, pp. 406–409.

[26] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Computer Vision, 2017 IEEE International Conference on*. IEEE, 2017, pp. 2980–2988.

[27] K. He, J. Sun, and X. Tang, "Guided image filtering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 6, pp. 1397–1409, 2013.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Computer Vision and Pattern Recognition, Proceedings of the IEEE Conference on*, 2016, pp. 770–778.

[29] D.-A. Huang, L.-W. Kang, Y.-C. F. Wang, and C.-W. Lin, "Self-learning based image decomposition with applications to single image denoising," *Multimedia, IEEE Transactions on*, vol. 16, no. 1, pp. 83–93, 2014.

[30] D.-A. Huang, L.-W. Kang, M.-C. Yang, C.-W. Lin, and Y.-C. F. Wang, "Context-aware single image rain removal," in *Multimedia and Expo, IEEE International Conference on*. IEEE, 2012, pp. 164–169.

[31] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *Computer Vision and Pattern Recognition, IEEE Conference on*, 2017.

[32] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wang, "A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2017.

[33] P.-M. Jodoin, S. Pierard, Y. Wang, and M. Van Droogenbroeck, "Overview and benchmarking of motion detection methods," *Background Modeling and Foreground Detection for Video Surveillance*, 2014.

[34] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *Image Processing, IEEE Transactions on*, vol. 21, no. 4, pp. 1742–1755, 2012.

[35] J.-H. Kim, C. Lee, J.-Y. Sim, and C.-S. Kim, "Single-image deraining using an adaptive nonlocal means filter," in *Image Processing, 20th IEEE International Conference on*. IEEE, 2013, pp. 914–917.

[36] J.-H. Kim, J.-Y. Sim, and C.-S. Kim, "Video deraining and desnowing using temporal correlation and low-rank matrix completion," *Image Processing, IEEE Transactions on*, vol. 24, no. 9, pp. 2658–2670, 2015.

[37] S. Lee, S. Yun, J.-H. Nam, C. S. Won, and S.-W. Jung, "A review on dark channel prior based image dehazing algorithms," *EURASIP Journal on Image and Video Processing*, vol. 2016, no. 1, p. 4, 2016.

[38] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," *arXiv preprint arXiv:1611.07709*, 2016.

[39] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain streak removal using layer priors," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2016, pp. 2736–2744.

[40] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*. Springer, 2014, pp. 740–755.

[41] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv preprint arXiv:1009.5055*, 2010.

[42] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1, pp. 503–528, 1989.

[43] P. Liu, J. Xu, J. Liu, and X. Tang, "Pixel based temporal analysis using chromatic property for removing rain from videos," *Computer and Information Science*, vol. 2, no. 1, p. 53, 2009.

[44] Y.-F. Liu, D.-W. Jaw, S.-C. Huang, and J.-N. Hwang, "Desnownet: Context-aware deep network for snow removal," *arXiv preprint arXiv:1708.04512*, 2017.

[45] Y. Luo, Y. Xu, and H. Ji, "Removing rain from a single image via discriminative sparse coding," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2015, pp. 3397–3405.

[46] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, no. Jan, pp. 19–60, 2010.

[47] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3397–3415, 1993.

[48] J. B. Mason, "Light attenuation in falling snow," Army Electronics Research and Development Command WSMR NM Atmospheric Science Lab, Tech. Rep., 1978.

[49] A. K. Moorthy and A. C. Bovik, "A two-step framework for constructing blind image quality indices," *IEEE Signal Processing Letters*, vol. 17, no. 5, pp. 513–516, 2010.

[50] W.-J. Park and K.-H. Lee, "Rain removal using kalman filter in video," in *Smart Manufacturing Application, ICSMA 2008 International Conference on*. IEEE, 2008, pp. 494–497.

[51] S.-C. Pei, Y.-T. Tsai, and C.-Y. Lee, "Removing rain and snow in a single image using saturation and visibility features," in *Multimedia and Expo Workshops, IEEE International Conference on*. IEEE, 2014, pp. 1–6.

[52] W. Ren, J. Tian, Z. Han, A. Chan, and Y. Tang, "Video desnowing and deraining based on matrix decomposition," in *Computer Vision and Pattern Recognition, IEEE Conference on*, July 2017.

[53] M. Roser and A. Geiger, "Video-based raindrop detection for improved image registration," in *Computer Vision Workshops, IEEE 12th International Conference on*. IEEE, 2009, pp. 570–577.

[54] V. Santhaseelan and V. K. Asari, "Utilizing local phase information to remove rain from video," *International Journal of Computer Vision*, vol. 112, no. 1, pp. 71–89, 2015.

[55] L. Shao, R. Yan, X. Li, and Y. Liu, "From heuristic optimization to dictionary learning: A review and comprehensive comparison of image denoising algorithms," *IEEE Transactions on Cybernetics*, vol. 44, no. 7, pp. 1001–1013, 2014.

[56] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *Image Processing, IEEE Transactions on*, vol. 15, no. 2, pp. 430–444, 2006.

[57] M. Shen and P. Xue, "A fast algorithm for rain detection and removal from videos," in *Multimedia and Expo, IEEE International Conference on*. IEEE, 2011, pp. 1–6.

[58] E. P. Shettle, "Models of aerosols, clouds, and precipitation for atmospheric propagation studies," in *In AGARD, Atmospheric Propagation in the UV, Visible, IR, and MM-Wave Region and Related Systems Aspects 14 p (SEE N90-21907 15-32)*, 1990.

[59] J. Shi *et al.*, "Good features to track," in *Computer Vision and Pattern Recognition, IEEE Conference on*. IEEE, 1994, pp. 593–600.

[60] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "Subsense: A universal change detection method with local adaptive sensitivity," *Image Processing, IEEE Transactions on*, vol. 24, no. 1, pp. 359–373, 2015.

[61] S. Starik and M. Werman, "Simulation of rain in videos," in *Texture Workshop, ICCV*, vol. 2, 2003, pp. 406–409.

[62] S.-H. Sun, S.-P. Fan, and Y.-C. F. Wang, "Exploiting image structural similarity for single image rain removal," in *Image Processing (ICIP), IEEE International Conference on*. IEEE, 2014, pp. 4482–4486.

[63] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, 2017, pp. 4278–4284.

[64] J. B. Tatum, "Stellar atmospheres," http://astrowww.phys.uvic.ca/~tatum/stellatm.html, 2011, [Online; posted 2017-08-08].

[65] A. K. Tripathi and S. Mukhopadhyay, "Rain rendering in videos," in *Computer Applications in Electrical Engineering Recent Advances, Int. Conf. on*, 2010, pp. 417–420.

[66] ——, "A probabilistic approach for detection and removal of rain from videos," *IETE Journal of Research*, vol. 57, no. 1, pp. 82–91, 2011.

[67] ——, "Meteorological approach for detection and removal of rain from videos," *IET Computer Vision*, vol. 7, no. 1, pp. 36–47, 2013.

[68] A. Tripathi and S. Mukhopadhyay, "Video post processing: low-latency spatiotemporal approach for detection and removal of rain," *IET Image Processing*, vol. 6, no. 2, pp. 181–196, 2012.

[69] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "Cdnet 2014: an expanded change detection benchmark dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 387–394.

[70] Y. Wang, S. Liu, C. Chen, and B. Zeng, "A hierarchical approach for rain or snow removing in a single color image," *Image Processing, IEEE Transactions on*, vol. 26, no. 8, pp. 3936–3950, 2017.

[71] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE transactions on*, vol. 13, no. 4, pp. 600–612, 2004.

[72] J. Xu, W. Zhao, P. Liu, and X. Tang, "An improved guidance image based method to remove rain and snow in a single image," *Computer and Information Science*, vol. 5, no. 3, p. 49, 2012.

[73] Y. Xu, J. Wen, L. Fei, and Z. Zhang, "Review of video and image defogging algorithms and related studies on image restoration and enhancement," *IEEE Access*, vol. 4, pp. 165–188, 2016.

[74] X. Xue, X. Jin, C. Zhang, and S. Goto, "Motion robust rain detection and removal from videos," in *International Workshop on Multimedia Signal Processing*. IEEE, 2012, pp. 170–174.

[75] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Deep joint rain detection and removal from a single image," in *Computer Vision and Pattern Recognition, IEEE Conference on*, July 2017.

[76] S. You, R. T. Tan, R. Kawakami, Y. Mukaigawa, and K. Ikeuchi, "Adherent raindrop modeling, detection and removal in video," *Pattern Analysis and Machine Intelligence, IEEE transactions on*, vol. 38, no. 9, pp. 1721–1733, 2016.

[77] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[78] H. Zhang and V. M. Patel, "Convolutional sparse and low-rank coding-based rain streak removal," in *Applications of Computer Vision, IEEE Winter Conference on*. IEEE, 2017, pp. 1259–1267.

[79] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *Computer Vision and Pattern Recognition, IEEE Conference on*, 2017.

[80] X. Zhang, H. Li, Y. Qi, W. K. Leow, and T. K. Ng, "Rain removal in video by combining temporal and chromatic properties," in *Multimedia and Expo, IEEE International Conference on*. IEEE, 2006, pp. 461–464.

[81] X. Zheng, Y. Liao, W. Guo, X. Fu, and X. Ding, "Single-image-based rain and snow removal using multi-guided filter," in *International Conference on Neural Information Processing*. Springer, 2013, pp. 258–265.

[82] M. Zhou, Z. Zhu, R. Deng, and S. Fang, "Rain detection and removal of sequential images," in *Chinese Control and Decision Conference*. IEEE, 2011, pp. 615–618.

[83] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Pattern Recognition. Proceedings of the 17th International Conference on*, vol. 2. IEEE, 2004, pp. 28–31.

Paper G

Learning to Remove Rain in Traffic Surveillance by Using Synthetic Data

Chris H. Bahnsen, David Vázquez, Antonio M. López, and Thomas B. Moeslund

The paper is published in the
Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP), in press, 2019.

The layout has been revised.

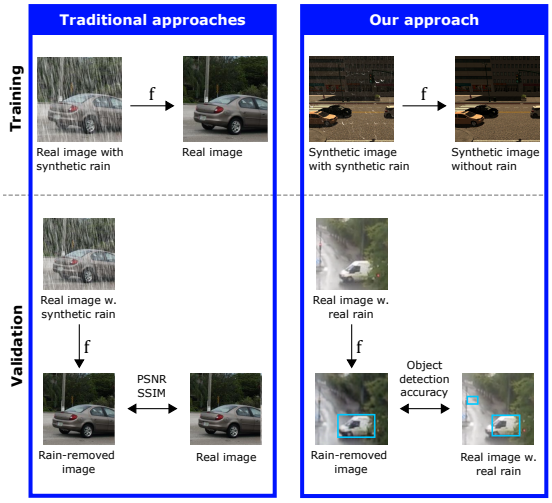


Fig. G.1: Proposed system-at-a-glance. As opposed to traditional methods, we use fully synthetic data for training rain removal algorithms. We validate on real-world images with real rain for which ground truth rain-free images do not exist, and we are thus unable to use traditional metrics such as SSIM or PSNR. Instead, we measure the accuracy of an object detection algorithm on both the original and rain-removed images. An effective rain removal algorithm should improve the visibility of foreground objects and thus increase the object detection accuracy.

Abstract

Rainfall is a problem in automated traffic surveillance. Rain streaks occlude the road users and degrade the overall visibility which in turn decrease object detection performance. One way of alleviating this is by artificially removing the rain from the images. This requires knowledge of corresponding rainy and rain-free images. Such images are often produced by overlaying synthetic rain on top of rain-free images. However, this method fails to incorporate the fact that rain fall in the entire three-dimensional volume of the scene. To overcome this, we introduce training data from the SYNTHIA virtual world that models rain streaks in the entirety of a scene. We train a conditional GAN for rain removal and apply it on traffic surveillance images from SYNTHIA and the RainSnow datasets. To measure the applicability of the rain-removed images in a traffic surveillance context, we run the YOLOv2 object detection algorithm on the original and rain-removed frames. The results on SYNTHIA show an 8% increase in detection accuracy compared to the original rain image. Interestingly, we find that high PSNR or SSIM scores do not imply good object detection performance.

1 Introduction

In computer vision-enabled traffic surveillance, one would hope for optimal conditions such as high visibility, few reflections, and good lighting conditions. This might be the case under daylight and overcast weather but is hardly representative of most real-life weather conditions. To name an example, the visibility of a scene might be impaired by the occurrence of precipitation such as rainfall and snowfall. The rain and snowfall are present in the images and videos as spatio-temporal streaks that might occlude foreground objects of interest. The accumulation of rain and snow streaks ultimately degrades the visibility of a scene [20] which render far-away objects hard to distinguish from the background. These rain and snow streaks may even adhere to the camera lens as quasi-static rain drops that remain for several seconds, effectively blurring a region of the image. The above mentioned properties of rain and snowfall have a detrimental effect on computer vision algorithms and the research community has therefore shown great interest to mitigate these effects. Since the first work by Hase *et al*. [8], many subsequent authors have proposed algorithms that tries to produce a realistic rain-removed image from a real-world rainy image.

When constructing an algorithm that artificially removes rain in an image or video, one would typically optimize for creating rain-removed images that resemble real-world images as much as possible. Typically, this is assessed by computing the Peak Signal-to-Noise-Ratio (PSNR) and the Structural Similarity Index (SSIM) [25] between the rain-removed image and the ground truth rain-free image. A high PSNR or SSIM score indicates that the source and target images are largely similar. The computation of these metrics, however, requires corresponding image pairs of rainy and rain-free images. For single-image rain removal, this requirement is usually met by overlaying artificial rain streaks on real-world images, typically by generating them in Adobe Photoshop or by using a collection of pre-rendered rain streaks [7]. A sample image is visible from the left part of Figure G.1.

Although the individual rain streaks may look realistic, the visual impression of the artificially produced rain image is less pleasing. Because the generated rain streaks are layered on top of the rain-free image, all rain streaks appear to be in the immediate foreground of the image. However, this fails to take into account that rain may fall in the entire three-dimensional volume of the scene and does not model the visibility degradation caused by the accumulation of rain drops.

The aim of this work is to create a single-image based rain removal algorithm that takes both the partial occlusions and the accumulation of rain into account. We accomplish this by introducing a new training dataset that consists of images from a purely synthetic, 3D-generated world. By using a

computer-generated 3D world, we can simulate raindrops in the entirety of the scene and not just in front of the camera. This enables us to mimic the rain streaks, the accumulation of rain, and the adhering of rain drops to the virtual camera lens. The concept is illustrated in Figure G.1.

Our contributions are the following:

1. To the best of our knowledge, we are the first to introduce fully synthetic training data for training and testing single-image based rain removal algorithms.

2. We train a rain removal algorithm using the data from above and compare with traditional approaches that use synthetic rain on top of real-world images. In order to assess the performance on real-world traffic surveillance images with real rain, we propose a new evaluation metric that assesses the performance of an object detection algorithm on the original and rain-removed frames. If effective, the rain-removed images should improve object detection performance.

3. The proposed evaluation metric is compared with the traditional PSNR and SSIM metrics to evaluate their usefulness in application-based rain removal.

2 Related Work

The first single-image based rain removal algorithm was proposed by Fu *et al*. [6] in 2011 and treated rain removal as a dictionary learning problem where the challenge is to decide if image patches belong to the rain component, R, or the background component, B. Relying on the assumption that rain drops are high-frequency (HF) oscillations occurring on top of a low-frequency (LF) background image, the bilateral filter is applied to the input image to separate it into a HF and a LF component. The Morphological Component Analysis technique [3] learns a dictionary of image patches from the HF image and rain streak patches are identified based on the assumption that they are brighter than other patches. The dictionary composition approach to rain removal was refined in subsequent works [1, 10, 14, 24].

An alternative approach was proposed by Chen *et al*. [2] that treats the separation of the rain image R from the background image B as a matrix decomposition problem. It is assumed that B has low total variation and that R patches are linearly dependent. Based on these assumptions, the Inexact Augmented Lagrange Multiplier is used to solve the constrained matrix decomposition problem. Subsequent works on matrix decomposition [12, 17] have imposed additional requirements on B and R such as low rank, sparsity, or mutual exclusivity.

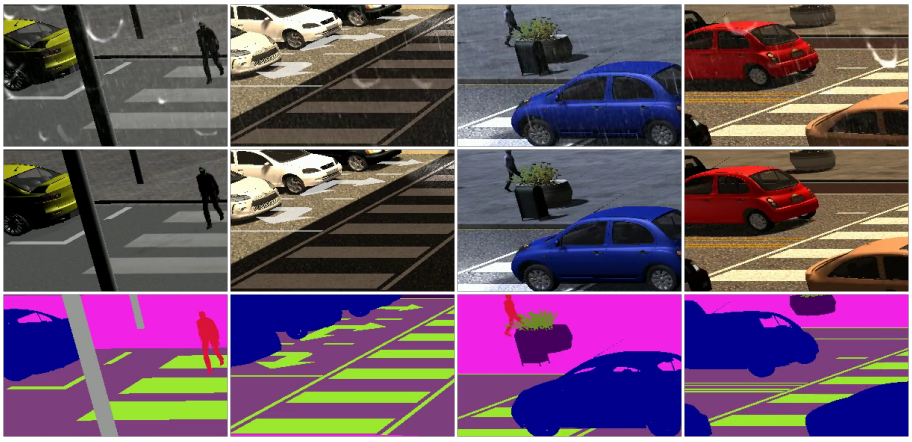


Fig. G.2: Synthetic images generated from SYNTHIA at four different locations in the virtual world. From top to bottom: rain image, no-rain image, ground truth segmented image. The images are cropped for viewing.

The Achilles heel of the mentioned dictionary component and matrix decomposition methods is that they solely rely on heuristically defined statistical properties to detect and remove the apparent rain. Real-world textures might not adhere to these statistical properties, however, and as a result, non-rain textures might be 'trapped' inside the rain component.

This problem is overcome by learning the appearance of rain streaks in an offline process that uses a collection of rain-free images overlaid with synthetic rain. Recent approaches use such images to train convolutional neural networks (CNNs) to remove rain from single images.

Fu *et al*. [4] combined a guided filter with a three-layer CNN to produce rain-free images. In the work to follow, the same author replaced the three-layer CNN with a much larger network containing residual connections [5].

A CNN containing dilated convolutions were used in [26] whereas Liu *et al*. [16] used a network based on the Inception V4 architecture [22]. As opposed to the works of Fu *et al*., both methods operate directly on the input image and may as such capture rain drops that are not included in the filtered HF image.

Conditional Generative Adversarial Networks The recent advent of generative adversarial networks (GANs) that are conditioned on the input image have made major breakthroughs in image-to-image translation [11]. A conditional GAN may be used to transfer an image in a specific domain to a corresponding image in another domain, for instance from rainy to rain-free images. Zhang *et al*. [27] modified the Pix2Pix framework by Isola *et al*. [11] by including the perceptual loss function by Johnson *et al*. [13] and trained the

conditional GAN on corresponding image pairs with and without synthetic rain.

3 Rain Removal Using Entirely Synthetic Data

In this section, we will describe our proposed rain removal framework. Like most other authors of rain removal algorithms, we want our network to be able to remove rain from real-world images with real rain, including the effects of rain streak occlusion and rain streak accumulation. The occluding effects of rain streaks might be modelled by imposing synthetic rain on real-world images but this approach cannot capture the effects arising from the accumulation of rain. In order to capture these effects, we propose to use fully synthetic training data generated from a computer-generated 3D world.

More specifically, we use renderings from the SYNTHIA virtual world [19] that capture four different road intersections as seen from an infrastructure-side traffic camera. The virtual world enables us to render two instances of the same sequence with the only difference that rain is falling in one instance but not in the other. Samples from the four sequences are shown in Figure G.2. In total, the four sequences comprise of 9572 frames.

A benefit of the SYNTHIA virtual world is that it enables the generation of corresponding segmented images that may be used directly as ground truth for semantic segmentation and object detection purposes. Footage from SYNTHIA has previously been used to successfully transfer images from summer to winter [9] or to transfer from SYNTHIA to the real-world Cityscapes dataset [28]. Based on these works, we therefore find it reasonable to learn the translation from rain images to no-rain images with the use of SYNTHIA.

Inspired by the recent work in image-to-image translation and domain adaption [9, 21], we use the conditional GAN architecture as the backbone of our rain removal framework.

3.1 Training the Conditional GAN-network

As point of departure, we take the rain removal algorithm from Zhang *et al*. [27], which consists of a conditional GAN-network, denoted as IDCGAN. We compare the IDCGAN with the state-of-the-art image-to-image translation framework Pix2PixHD [23].

The discriminator of the IDCGAN-network uses a five-layer convolutional structure similar to the original Pix2Pix-network [11] whereas the generator uses a fully convolutional network with skip connections, the U-net. The generator architecture is different from the Pix2Pix-network in two ways:

1. The depth of the U-net is down from eight to six convolutional layers.

Name	Training data
IDCGAN-Real-Syn	Real images with synthetic rain
IDCGAN-Syn-Syn	SYNTHIA rain images, SYNTHIA no-rain images
Pix2PixHD-Real-Syn	Real images with synthetic rain
Pix2PixHD-Syn-Syn	SYNTHIA rain images, SYNTHIA no-rain images
Pix2PixHD-Combined	SYNTHIA rain images, SYNTHIA no-rain images + real images with synthetic rain

Table G.1: Overview of the trained conditional GANs for rain removal. The training of IDCGAN-Real-Synth is equivalent to the original work of Zhang *et al*. [27].

2. The skip-connections are adding the tensors instead of concatenating (joining) them.

The Pix2PixHD network is an improved version of Pix2Pix that enables the generation of more realistic, high-resolution images.

As training set, we use the aforementioned SYNTHIA dataset with 9572 corresponding image pairs. As representative of a dataset with real images and synthetic rain, the 700 training images from Zhang *et al*. [27] are used. The IDCGAN and Pix2PixHD networks are trained separately with the images of Zhang *et al*. and the SYNTHIA training images. Furthermore, we use a combination of the two datasets to train the Pix2PixHD network. An overview of the resulting five trained networks is found in Table G.1. In order to make the training feasible on a 11 GB GPU, the training images are scaled down to a maximum resolution of 720 x 480 pixels. Otherwise, we use the default parameter settings for training the networks.

4 Assessing The Rain Removal Quality

As mentioned in the introduction, the classical approach of measuring the quality of the rain-removed image is to apply a rain removal algorithm on a rain-free image with overlaid synthetic rain and calculate the PSNR and SSIM between the resulting image with the corresponding rain free-image. In a traffic surveillance context, it appears that the overlaid synthetic rain hardly resembles real-world rain. As such, there is no guarantee that a rain removal algorithm receiving high PSNR and SSIM scores on synthetic rain will translate well to real-world rain in a traffic surveillance image.

We therefore propose a new evaluation metric that measures the ability of an object detection algorithm to detect objects in the original and the rain-removed frames. If the rain removal algorithm has succeeded, it has created a

rain-removed image that resembles a true rain-free image. This means that the occlusion and visibility degradation originating from the rain streaks should be largely eliminated, creating an image in which objects are easier to detect. Instead of requiring the overlay of synthetic rain on rain-free images, this metric requires the annotation of bounding boxes around objects of interest. We find such sequences in the RainSnow dataset[1] that contains 2200 annotated frames in a traffic surveillance context, taken from seven different traffic intersections. The dataset features a variety of challenging conditions such as rain, snow, low light, and reflections.

As object detection benchmark, we choose the state-of-the-art You Only Look Once algorithm (YOLOv2) [18]. YOLOv2 is chosen due to good detection performance and superior speed which is especially important in real-time traffic surveillance. The improvement in detection performance is assessed by:

1. Running pre-trained YOLOv2 on the original, rainy images of the SYNTHIA dataset.

2. Removing rain with the networks listed in Table G.1 and running pre-trained YOLOv2 on the rain-removed images.

3. Measuring the detection accuracy of 1) and 2) by using the COCO API [15] and calculating the relative difference.

We also measure the improvement in detection performance on the RainSnow dataset by following the above steps, replacing SYNTHIA with RainSnow.

5 Experimental Results

We have experimented with several hyper-parameter settings for YOLOv2 and found the best results by setting the detection threshold, hierarchical threshold, and the non-maximum suppression threshold to 0.1, 0.1, and 0.3, respectively. As detection metric, we use average precision (AP) over intersection-over-union (IOU) ratios from .5 to .95 with intervals of .05, denoted as AP[.5:.05:.95], and average precision at IOU=0.5, denoted as AP[.5].

5.1 Removing Rain From SYNTHIA Training Data

We start by measuring the ability to remove rain from the SYNTHIA data. This is a peculiar case as the Syn-Syn networks have seen the data during the training phase and we are thus unable to judge whether these algorithms generalize well. It does, however, give the opportunity to assess feasibility of

[1]The dataset will be publicly available with the camera-ready version of this paper.

Rain removal method	SSIM	PSNR	YOLOv2	
Original rain image	-	-	*.025*	*.072*
Original no-rain image	-	-	38.4	23.6
IDCGAN-Real-Syn	.610	65.3	8.78	2.02
IDCGAN-Syn-Syn	.873	80.8	1.51	-7.15
Pix2PixHD-Real-Syn	.646	69.3	-32.0	-36.2
Pix2PixHD-Syn-Syn	.767	75.9	8.07	7.35
Pix2PixHD-Combined	.640	70.4	-32.7	-34.5

Table G.2: Results on the SYNTHIA dataset. YOLOv2 results in AP[.5:.05:.95], AP[.5]. Absolute values are reported for the original rain image in italics, whereas other YOLOv2 results are relative to this baseline, shown in percentages. The original no-rain images are used as reference for computing SSIM and PSNR scores.

the rain removal algorithms in a best-case scenario and relate SSIM/PSNR scores and object detection performance, reported in Table G.2.

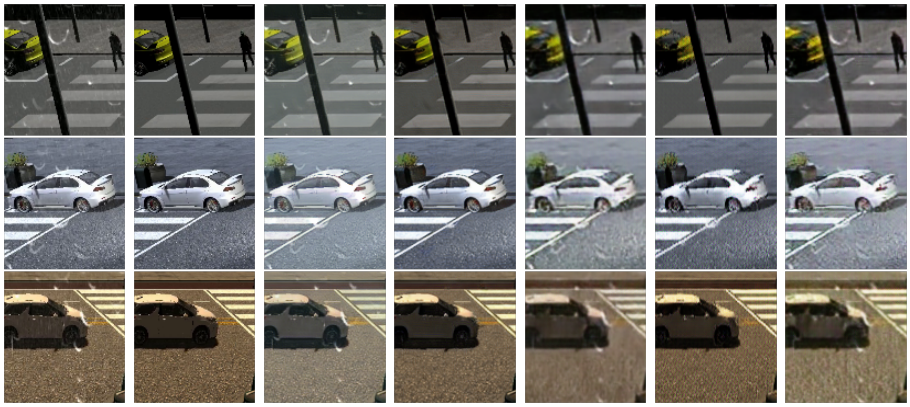
The detection results of Table G.2 show that only two rain removal algorithms, IDCGAN-Real-Syn and Pix2PixHD-Syn-Syn, improve detection performance compared to the original rain images, but neither of two algorithms come close at the detection performance of the ground truth no-rain images. This is remarkable given the fact that Pix2PixHD-Syn-Syn has seen these images during training.

Interestingly, the SSIM scores of the two rain removal algorithms show little correspondence with the detection results. The IDCGAN-Real-Syn network is receiving the lowest SSIM score but shows good detection performance whereas the IDCGAN-Syn-Syn network is receiving the highest SSIM score but fails to consistently improve the detection results.

Example images from the SYNTHIA data are shown in Figure G.3. The networks trained solely on the SYNTHIA data are able to remove the majority of rain from the image with IDCGAN-Syn-Syn leaving the best visual impression, whereas the Pix2PixHD-Syn-Syn network suffers from checkerboard artifacts in the reconstructed textures.

5.2 Removing Rain From RainSnow sequences

Sample images from running the rain removal algorithms on the RainSnow dataset are shown in Figure G.4 whereas detection results from running YOLOv2 on the rain-removed images are found in Table G.3. The detection results show marginal improvements on the networks trained on real images with synthetic rain (Real-Syn and Combined), whereas networks trained on only synthetic data (Syn-Syn) deteriorate the detection results. If we look at the visual examples from Figure G.4, the rain-removed images of IDCGAN-Syn-Syn have strange artifacts and do not seem to lie within the domain of



(a) Original rain image | **(b)** Original no-rain image | **(c)** IDCGAN-Real-Syn | **(d)** IDCGAN-Synth-Syn | **(e)** Pix2PixHD-Real-Syn | **(f)** Pix2PixHD-Synth-Syn | **(g)** Pix2PixHD-Combined

Fig. G.3: Rain-removal results on the SYNTHIA dataset. Each column represents the results of a rain removal algorithm on the original rain image.

Rain removal method	YOLOv2	
Original rain image	*.034*	*.070*
IDCGAN-Real-Syn	1.17	0.54
IDCGAN-Syn-Syn	-47.9	-42.9
Pix2PixHD-Real-Syn	-1.08	3.87
Pix2PixHD-Syn-Syn	-14.5	-5.05
Pix2PixHD-Combined	-2.43	3.19

Table G.3: Detection results on the RainSnow dataset. Results in AP[.5:.05:.95], AP[.5]. Absolute values are reported for the original rain image in italics, whereas other YOLOv2 results are relative to this baseline, shown in percentages.

visual images, whereas the images from the Pix2PixHD-Syn-Syn network appear to lie closer to the visual domain. The latter network even attempts to remove the rain drops from the lower image at Figure G.4 and removes both the large rain streak and the reflections from the cars from the top image.

In general, however, the visual results also reveal plenty of room for improvement for all rain removal algorithms. As an example, the rain streaks on the top image and the rain drops on the lower image are not efficiently removed by any algorithm.

(a) Original rain image **(b)** IDCGAN-Real-Syn **(c)** IDCGAN-Synth-Syn **(d)** Pix2PixHD-Real-Syn **(e)** Pix2PixHD-Synth-Syn **(f)** Pix2PixHD-Combined

Fig. G.4: Rain-removal results on RainSnow dataset. Each column represents the results of a rain removal algorithm on the original rain image.

5.3 Domain Transfer Results

We find that the IDCGAN and Pix2PixHD networks behave inconsistently when tested on sequences that are dissimilar from their training data. On SYNTHIA, IDCGAN-Real-Syn improves the detection results, whereas Pix2PixHD-Real-Syn fails to do so, even if providing a higher SSIM score. The results are reversed on RainSnow sequences, with IDCGAN-Syn-Syn providing much worse detection results that Pix2PixHD-Syn-Syn. No obvious explanation of this behaviour exists and further experiments are needed in order to find and understand the most suitable network for domain transfer in rain removal.

6 Conclusions

We have investigated the use of fully synthetic data from the SYNTHIA virtual world to train a GAN-based, single-image rain removal algorithm. Using the fully synthetic data, we find that there is a considerable gap between detection scores on the rain-removed images from the best-performing rain removal algorithm and detection scores on the ground truth images with no rain. Furthermore, we found no correlation between SSIM or PSNR scores and detection performance, questioning the usefulness of these metrics for application-based rain removal.

Removing rain on real-world traffic surveillance imagery is hard and the evaluated rain-removal only results in marginal improvements in detection performance, if any. Using fully synthetic data for training allows the removal

of some rain streaks that were not captured by networks trained with only synthetic rain. There exists, however, a domain gap between the synthetic data and the real-world sequences. Future work should address this by including more diverse synthetic data and more variability in real-world synthetic rain. One could also investigate the use of recurrent neural networks to incorporate temporal information from the SYNTHIA dataset. In traffic surveillance where the rain removal could be an intermediate step in achieving good detection performance, it might even be beneficial to use the synthetic rain images for training a classifier and skip the rain removal step altogether.

References

[1] D.-Y. Chen, C.-C. Chen, and L.-W. Kang, "Visual depth guided color image rain streaks removal using sparse coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 8, pp. 1430–1455, 2014.

[2] Y.-L. Chen and C.-T. Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," in *IEEE International Conference on Computer Vision, Proceedings of the*, 2013, pp. 1968–1975.

[3] M. J. Fadili, J.-L. Starck, J. Bobin, and Y. Moudden, "Image decomposition and separation using sparse representations: an overview," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 983–994, 2010.

[4] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *Image Processing, IEEE Transactions on*, vol. 26, no. 6, pp. 2944–2956, 2017.

[5] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *Computer Vision and Pattern Recognition, IEEE Conference on*, July 2017.

[6] Y.-H. Fu, L.-W. Kang, C.-W. Lin, and C.-T. Hsu, "Single-frame-based rain removal via image decomposition," in *Acoustics, Speech and Signal Processing, IEEE International Conference on*. IEEE, 2011, pp. 1453–1456.

[7] K. Garg and S. K. Nayar, "Photorealistic rendering of rain streaks," in *Graphics, ACM Transactions on*, vol. 25, no. 3. ACM, 2006, pp. 996–1002.

[8] H. Hase, K. Miyake, and M. Yoneda, "Real-time snowfall noise elimination," in *Image Processing, International Conference on*, vol. 2. IEEE, 1999, pp. 406–409.

[9] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," *arXiv preprint arXiv:1711.03213*, 2017.

[10] D.-A. Huang, L.-W. Kang, Y.-C. F. Wang, and C.-W. Lin, "Self-learning based image decomposition with applications to single image denoising," *Multimedia, IEEE Transactions on*, vol. 16, no. 1, pp. 83–93, 2014.

[11] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint*, 2017.

[12] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wang, "A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2017.

[13] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*. Springer, 2016, pp. 694–711.

[14] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *Image Processing, IEEE Transactions on*, vol. 21, no. 4, pp. 1742–1755, 2012.

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*. Springer, 2014, pp. 740–755.

[16] Y.-F. Liu, D.-W. Jaw, S.-C. Huang, and J.-N. Hwang, "Desnownet: Context-aware deep network for snow removal," *arXiv preprint arXiv:1708.04512*, 2017.

[17] Y. Luo, Y. Xu, and H. Ji, "Removing rain from a single image via discriminative sparse coding," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2015, pp. 3397–3405.

[18] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint*, 2017.

[19] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3234–3243.

[20] E. P. Shettle, "Models of aerosols, clouds, and precipitation for atmospheric propagation studies," in *In AGARD, Atmospheric Propagation in the UV, Visible, IR, and MM-Wave Region and Related Systems Aspects 14 p (SEE N90-21907 15-32)*, 1990.

[21] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *CVPR*, vol. 2, no. 4, 2017, p. 5.

[22] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, 2017, pp. 4278–4284.

[23] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," *arXiv preprint arXiv:1711.11585*, 2017.

[24] Y. Wang, S. Liu, C. Chen, and B. Zeng, "A hierarchical approach for rain or snow removing in a single color image," *Image Processing, IEEE Transactions on*, vol. 26, no. 8, pp. 3936–3950, 2017.

[25] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE transactions on*, vol. 13, no. 4, pp. 600–612, 2004.

[26] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Deep joint rain detection and removal from a single image," in *Computer Vision and Pattern Recognition, IEEE Conference on*, July 2017.

[27] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *Computer Vision and Pattern Recognition, IEEE Conference on*, 2017.

[28] Y. Zhang, P. David, and B. Gong, "Curriculum domain adaptation for semantic segmentation of urban scenes," in *The IEEE International Conference on Computer Vision (ICCV)*, vol. 2, no. 5, 2017, p. 6.

Paper H

Detecting Road User Actions in Traffic Intersections Using RGB and Thermal Video

Chris Bahnsen and Thomas B. Moeslund

The paper has been published in the
Proceedings of the 2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2015.

The layout has been revised.

Abstract

This paper investigates the development of a watch-dog system that detects a subset of road user actions in traffic intersections. Footage of the intersections is captured with RGB and thermal cameras to ensure that the road is visible round-the-clock even in difficult weather conditions. The watch-dog system consists of several, cascaded detectors which are capable of detecting specific road user actions, such as Right Turning Vehicles, Left Turning Vehicles, and Straight Going Cyclists. Experimental results on 4 hours of video from 3 different intersections show good performance and a precision above 0.93 when detecting turning vehicles. The use of both RGB and thermal video generally results in better performance, providing overall stability when observing the road.

1 Introduction

It is the goal of the European Commission to cut the number of road deaths by 50 % in 2020 and diminish the number almost entirely by 2050 [3]. In order to reach these goals, not only the security of the vehicles must be enhanced but also the layout of the roads must change to enhance safety. Historically, road layouts have been changed to the better based on previous knowledge of road fatalities and deaths. This means that traffic researches and designers must wait for accidents to happen in order to improve the layout of the road.

In surrogate safety analysis, however, it is sufficient to measure the number of accidents that almost happened. The foundation behind surrogate analysis is the existence of a continuous relationship between the levels of severity of an accident and their corresponding frequencies. For instance, it is assumed that slight injuries occur more frequently than severe injuries and thus one may form a safety pyramid [6] where the fatal injuries resist at the very upper parts of the pyramid (severe, low frequency) and normal traffic fill up the bottom parts of the pyramid (normal, high frequency). By counting the number of near-accidents where a critical interaction between road users nearly happened, one achieves a *surrogate measure* of the number of more severe, fatal interactions [14]. Recently, this rationale has been taken even further by indicating that less-severe, normal traffic interactions enables traffic researches to monitor the safety level [11], [15]. This enables a rapid safety analysis of roads from data over weeks instead of years.

Special attention is needed in improving the safety of vulnerable road users (VRU). VRUs is defined as pedestrians, elderly, disabled persons, cyclists, and riders of powered two-wheelers (mopeds and motorcycles). Compared to the total number of traffic accidents, VRUs account for a disproportionately high number of road fatalities and injuries. In 2013, according to the European Commission [3] more than 14.000 VRUs were killed in the European Union. It

is the long-time goal of this project to enable traffic researches to improve the safety of VRUs by gaining knowledge of the accident causations. In this work, we are laying the foundation by studying specific movements of selected road users at intersections.

1.1 Monitoring road users

We have to study the roads in order to understand the frequency and nature of accidents and near-accidents. Manually monitoring the roads is tedious and inflexible and does not allow for a larger understanding of accident causation. A more flexible approach is to record the roads with a camera and watch the footage off-line. This allows for the reconstruction of critical events but still presents the user with a tremendous amount of data. The optimal solution to this problem is to design a system that automatically detects and tracks the road users from the recorded video data. From these tracks, traffic analysts can define heuristics that determine the interactions between the road users and on a higher level, the safety of a particular road.

However, the detection and tracking of objects in unconstrained scenes is still an unsolved problem. State-of-the art tracking systems are usually evaluated at 2-minute intervals under static weather conditions and does not perform well under occlusion, clutter, and illumination changes. No tracker is currently capable of detecting and tracking objects round-the-clock in unconstrained scenarios. Smeulders et al [13] provides a good review and performance evaluation of recent trackers.

Jackson et al [7] have developed an open-source toolbox for traffic video analysis which forms the base line for surrogate traffic analysis such as the work of [12]. However, as with general tracking algorithms, the length of the dataset is short and the video data is captured under good weather conditions [10]. The toolbox builds upon the popular KLT-tracker [1] which needs an additional grouping of tracked features to convert a number of tracked points to a number of tracked objects. The grouping is often ambiguous - is it a single bicycle or a cluster of cyclists waiting at the stop line? Is it a truck with a trailer - or two separate vehicles?

1.2 Reducing the amount of video

Because tracking still remains an unsolved problem, we acknowledge that there is a need for a human-in-the-loop to assess the nature and severity of the events between road users. However, we may design a system that reduces the amount of video data to manually assess. Such a *watch dog* should not necessary track all road users at all times but instead detect whenever there are situations that need further investigation - and whether there are periods of time when nothing of interest occurs. In this work, we will refrain from

detecting interactions between road users but instead study the individual actions of the road users and obtain a reliable detection. Once these detections are achieved, one may obtain interactions by combining the detections. We build upon the ideas introduced by Madsen et al [9]. In this work, we introduce a thorough evaluation of the individual detectors on novel datasets in both the RGB and thermal domains. Furthermore, we explain the algorithmic framework behind the detectors and how they are enhanced to work in both domains.

The issue of observing the road through a camera is treated in Section 2. The proposed watch dog that operates on the video data is presented in Section 3. Experiments are discussed in Section 4 and concluding remarks are presented in Section 5.

2 Observing the road

Even the best tracking systems are only as good as the data they process. We want to detect the road users round-the-clock in all weather conditions which means that the road should be observable in almost any condition by looking at the recordings provided. Traditional surveying techniques employ one or multiple visible light (RGB) cameras to monitor the intersection [8]. While this works well under good weather conditions, the video data still stuffer from varying shadows and very sparse information during the night. Thermal cameras, on the other hand, capture the radiated heat from objects and are thus not sensitive to changes in the environment as long as the object of interest has a different temperature from the background. For a survey of thermal cameras, refer to [5]. However, the thermal modality is poor on features which makes it much harder to discriminate between objects, recover identities after occlusion, or classify road users. Together though, RGB and thermal cameras supplement each other and extend the visibility of the road. In this work, we use a joint configuration of a RGB and thermal camera to monitor road intersections. See Figure H.1 for a comparison of the two modalities.

3 Watch-dog system

Because our system should be able to function as a watch-dog to a human operator, robustness to changes in the environment is more important than the ability to perfectly detect and track road users. In order to make the watch-dog robust, we tailor the system to perform a number of specific tasks in certain areas of interest. We use the geometry of the intersection to infer specific patterns that road users must take to complete an action. For instance, if we want to analyse a vehicle doing a right-turn at an intersection we know

(a) RGB

(b) Thermal

(c) RGB

(d) Thermal

Fig. H.1: (a), (b): RGB and thermal images of an intersection at dusk. In the RGB modality, the headlights of the cars passing by dominates most of the road. In the thermal modality, the cars are fully visible. (c), (d): RGB and thermal images of an intersection in full sunlight. In the thermal image, the car on the right is barely visible due to the heated asphalt whereas the biker on the upper pedestrian crossing stands out. The RGB image is fully visible.

that the vehicle must (1) enter the intersection, (2) perform a right turn, and (3) exit the intersection. These tasks may be solved in succession:

1. **Detect presence:** Detect if an object is present at the chosen entry point of the intersection. If the size of the object fulfils the criteria of the vehicle type, proceed to step 2. Otherwise, discard the object.

2. **Detect movement:** Detect if the object of interest is turning right, e.g. if there is movement in a certain direction in a predefined area of the intersection. If the movement is sufficient, proceed to step 3.

3. **Detect presence:** Detect if the object is present at the chosen exit point of the intersection by applying the method of step 1.

We assume that a vehicle has made a right turn if the three tasks are completed in succession. If not, the vehicle is doing something else - which another detector may detect.

In this specific context, we create the foundation to detect near-conflicts between vehicles and cyclists at urban signalized traffic intersections. In order to do so, we want to detect right turning vehicles, left turning vehicles, and straight going cyclists. The three detectors all consists of a chained combination of the two basic tasks; *detecting presence* and *detecting movement* which are further described in the following.

3.1 Detecting presence

When detecting presence, we want to detect if a road user is present or not at a given region of interest in the image. This is obtained via a background subtraction technique applied to the specific region of interest (ROI). We use a background subtraction technique based on reference images which are updated according to the routine described below:

1. Perform Canny edge detection [2] on current image and obtain edge image.

2. Subtract edge image from background edge image.

3. Filter noise.

4. Find pixel sum of remaining edges. If sum is above threshold, the detector is triggered.

5. Update background if the following criteria are satisfied:

   (a) Motion between current and previous frame is below 10 % of threshold for τ_1 concurrent frames.

(b) Pixel sum is below 80 % of threshold, and background has not been updated for τ_2 consecutive frames.

The routine above is applied independently on both the RGB and thermal modality. The threshold is found experimentally for each intersection and modality and is higher when detecting vehicles than detecting bicycles due to the difference in size of these road users.

3.2 Detecting movement

Estimation of the movement in a ROI of the video is obtained by using the two-frame dense motion estimation of Farneback [4] with the following procedure:

1. Calculate the dense optical flow of the ROI.
2. Count number of flow vectors of certain magnitude inside a chosen flow range.
3. Threshold vector count and update confidence measure.

The flow range mentioned in step 2 is chosen to only detect movement in the preferred range of the detector. For instance, we only want do detect movement from left to right when detecting right turning vehicles.

3.3 Chaining actions

It is of special interest of the traffic researchers to know whenever a road user is stationary in certain areas of the intersection. Therefore, we combine the tasks of detecting presence and movement into a third detector, the stationary object detector. The stationary object detector is triggered whenever something is present within the ROI and there is no or little movement, or flow, inside the ROI.

As described at the beginning of Section 3 we define events inside the intersection by chaining sequential actions. By tailoring the detectors for specific needs we focus the overall generic tracking problem to solve a very constrained problem at hand. Other problems, for instance right turning cyclists, might be solved by building another chained set of detectors. The task of detecting right turning vehicles is performed by the use of five detectors; two presence detectors, abbreviated E, two movement detectors (F), and one stationary detector (S). The number of detectors used for detecting left and right turning vehicles, and straight going cyclists is listed in Table H.1.

A vehicle is detected as a right turning candidate whenever it enters the entry point of the intersection which is laid out in the ROI of detector E1 (Figure H.2a). Whenever detector E1 is triggered, the movement detector

	RTV	LTV	SGC
Detecting presence (E)	2	2	3
Detecting movement (F)	2	4	1
Stationary object (S)	1	0	0

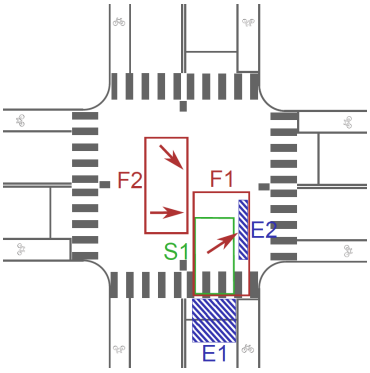
Table H.1: Detector types, and their shorthand notation, used when detecting Right Turning Vehicles (RTV), Left Turning Vehicles (LTV), and Straight Going Cyclists (SGC).

F1 and the stationary detector S1 are activated. The detector F1 looks for movement in the direction of the arrow (see Fig. H.2a) and detector S1 detects if the vehicle has stopped. If F1 has detected that the vehicle is turning, the detector E2 is activated to judge if the vehicle enters the conflict zone which concludes the detection. If S1 is activated, we assume that the vehicle has stopped in the middle of the intersection and is possibly awaiting clearance to turn. In this case, we let the other detectors stay open a little longer to detect an eventual turn of the vehicle. If no action occur in the detectors E2, F2, and S1, they are deactivated after a short duration of time. The detector F2 is used to filter out false positives, for instance vehicles going from left to right in the intersection. An activity diagram explaining the work-flow of the Right Turning Vehicle (RTV) detector is shown in Figure H.3. The RTV detector is shown on an intersection prototype in Figure H.2a and in an actual configuration in Figure H.2b.

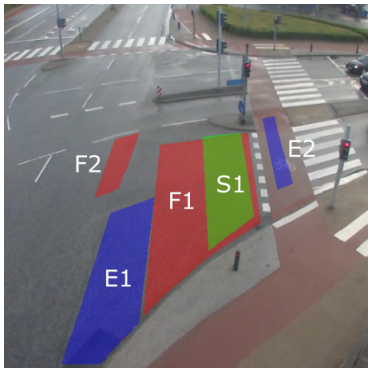
The Left Turning Vehicle (LTV) detectors and Straight Going Cyclist (SGC) detectors work similarly to the RTV detector. In the LTV detector, the stationary detector is discarded and the area of the presence detector (E1) is moved further into the intersection. Two movement detectors (F3, F4) have been added to filter out false detections from vehicles from other directions, complementing the F2 of the RTV detector. The proposed layout of the LTV detector is shown in Figure H.2c. The SGC detector adds one presence detector to help filter pedestrians and cars from cyclists. It discards the detectors F2, F3, and F4 as they have shown to be of little use in this specific case. The SGC detector prototype is seen in Figure H.2d. A straight going cyclist is detected if the detector E3 is activated in a chain of actions.

3.4 Fusing modalities

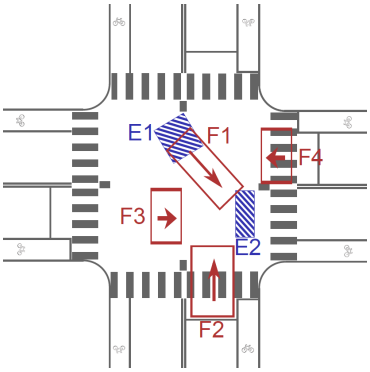
The video data of the intersections is captured by both a conventional RGB and a thermal camera. In this experiment, we synchronize the two modalities and run the detectors on each modality concurrently. Each underlying detector, i.e. the presence and movement detector, operates on both a RGB and a thermal image. For each modality, the detector outputs a confidence value between 0 and 1. An individual detector is triggered if the confidence is above 0.5. A



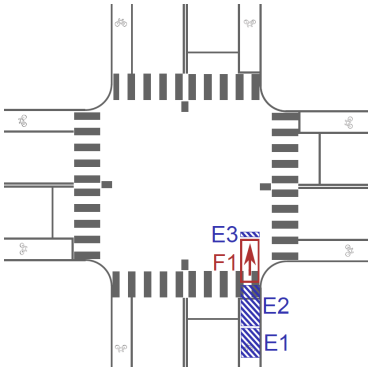
(a) Right Turning Vehicle (RTV)



(b) RTV on intersection C(1)



(c) Left Turning Vehicle (LTV)



(d) Straight Going Cyclist (SGC)

Fig. H.2: RTV, LTV, and SGC detectors on intersection prototypes

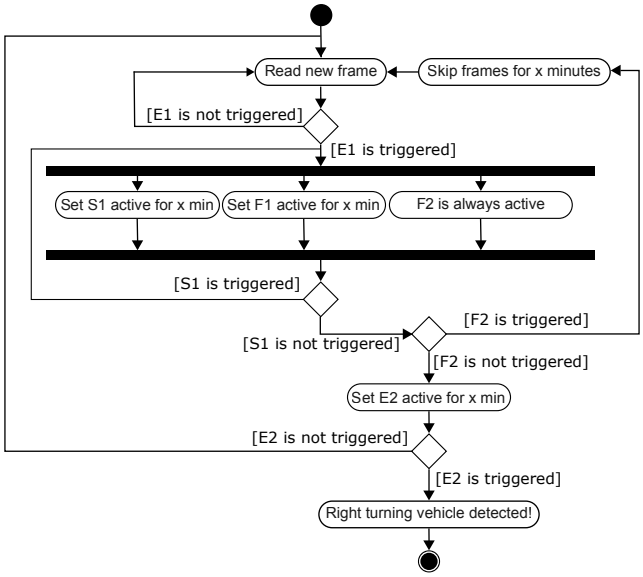


Fig. H.3: Activity diagram of the Right Turning Vehicle (RTV) detector.

multi-modal detector must have an averaged confidence value above 0.5 to be triggered.

4 Experimental results

The RTV, LTV, and SGC detectors are evaluated at three different intersections located in the Danish cities of Aalborg (A, B) and Viborg (C). The duration of the evaluated video data is four hours in total. The data is captured in the morning peak hour to capture as much traffic as possible and thus challenge the algorithms. The conditions of the evaluated intersections are listed in Table H.2. Samples from the intersections are shown in Figure H.4.

Intersection	Time	Weather	Temperature
A(1)	07:00 - 08:00	Sunny	13 °C
A(2)	07:00 - 08:00	Overcast	15 °C
B(1)	07:00 - 08:00	Rain	12 °C
C(1)	07:00 - 08:00	Overcast	13 °C

Table H.2: Conditions of the evaluated video data. Video A(2) is showing the same intersection as A(1), four days later.

For each of the locations, right turning vehicles, left turning vehicles, and

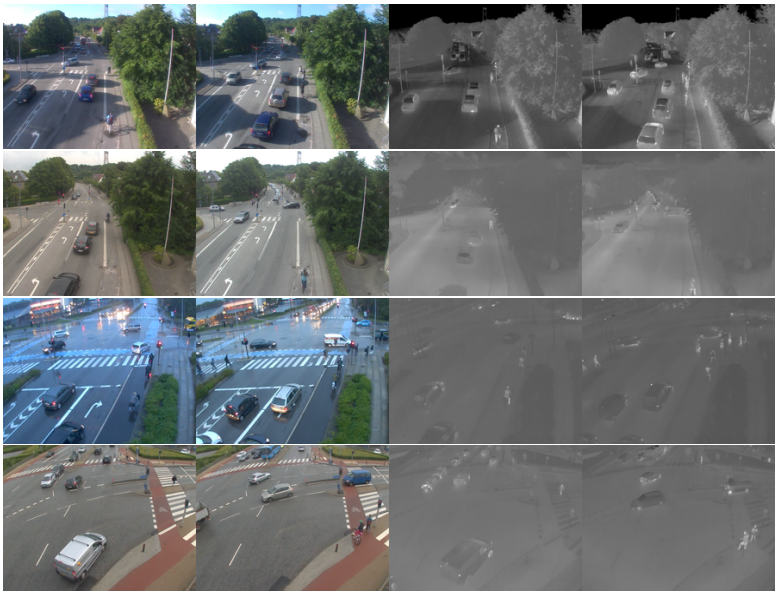


Fig. H.4: Snapshots of the intersections used in the experiments. For each intersection, two frames are shown in both the RGB and thermal modalities. From top to bottom; A(1), A(2), B(1), C(1).

straight going cyclists have been annotated manually and assigned a time stamp which corresponds to the entry of the vehicle or cyclist in the final presence detector (E2/E3) of the RTV, LTV, and SGC detectors. The detectors are fitted to each of the intersections using the first ten minutes of video. For sequence A(1) and A(2), the same settings are used. A detection is considered a true positive if its time stamp is within ± 2 seconds of the nearest ground truth time stamp. Detections and the corresponding ground truths can only be associated once, i.e. only one of multiple detections may be marked as a true positive if they all correspond to the same ground truth label. The results of the experimental evaluation are listed in Table H.3. The detectors are evaluated on the RGB and thermal modalities both separately and combined.

Overall, the results show good performance of the RTV and LTV detectors, resulting in a precision of 0.94–1.00 and a recall of 0.80–0.97 when combining both modalities (RGBT). The SGC detector performs poorer than the RTV and SGC in the four sequences, most notably in the RGB modality. The poorer performance of the cyclist detection is possibly due to occlusion and the case that cyclists riding side-by-side are detected as a single cyclist. Cyclists are more distinguished in the thermal modality which is reflected by higher precision rates than the corresponding RGB detections.

In 15 out of 24 cases (precision+recall), the detectors operating on RGBT perform better than or equal to the best performing single modality. In the

remaining 9, the performance is better in a single modality. However, in these cases, the RGBT is trailing behind the best performing modality by typically 0.01–0.03, even if the other single modality performs considerably worse.

5 Conclusions

This work presented a system that detects right and left turning vehicles, and straight going cyclists in signalized intersections by using RGB and thermal video data. It does so by chaining the output of two fundamental detectors which detects presence and movement. The spatial constraints of the intersections are used to create chains of actions that classifies a road user. The detectors are evaluated on a total of four hours of data from three different intersections. The results are promising and shows that the combination of RGB and thermal video may lead to a more stable detection of the road users in real-life, long-term traffic video.

Future work includes a more sophisticated fusion of the modalities by using contextual information to create a confidence measure reflecting the reliability of a modality. Furthermore, the detections will be combined to produce an estimate of the interactions between road users at the selected intersections.

Acknowledgements

The authors thank Tanja Kidmann Osmann Madsen for acquiring the data as well as providing the ground truth. This research was supported by a grant from the European Commission under the Horizon 2020-programme, H2020-EU.3.4.

References

[1] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[2] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 679–698, 1986.

[3] European Commission, "White paper roadmap to a single european transport area towards a competitive and resource efficient transport system," *COM (2011)*, vol. 144, 2011.

[4] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image Analysis*. Springer, 2003, pp. 363–370.

[5] R. Gade and T. B. Moeslund, "Thermal cameras and applications: A survey," *Machine vision and applications*, vol. 25, no. 1, pp. 245–262, 2014.

Intersection A(1) (1 hour)

	TP			FP			FN			Precision			Recall		
	SGC	RTV	LTV	SGC	RTV	LTV	SGC	RTV	LTV	SGC	RTV	LTV	SGC	RTV	LTV
RGB	**103**	61	65	420	57	5	**43**	61	12	0.20	0.52	0.93	**0.71**	0.50	0.84
T	102	92	50	31	**2**	**3**	44	30	27	0.77	**0.98**	0.94	0.70	0.75	0.65
RGBT	**103**	**97**	**71**	**25**	3	**3**	**43**	**25**	**6**	**0.80**	0.97	**0.96**	**0.71**	**0.80**	**0.92**

Number of positives: SGC 146, RTV 122, LTV 77

Intersection A(2) (1 hour)

	TP			FP			FN			Precision			Recall		
	SGC	RTV	LTV	SGC	RTV	LTV	SGC	RTV	LTV	SGC	RTV	LTV	SGC	RTV	LTV
RGB	92	**108**	**90**	43	2	9	49	**12**	**3**	0.68	0.98	0.91	0.65	**0.90**	**0.97**
T	91	102	15	16	2	**1**	50	18	78	0.85	0.98	0.94	0.65	0.85	0.16
RGBT	**97**	**108**	83	**11**	**0**	3	**44**	**12**	10	**0.90**	**1.00**	**0.97**	**0.69**	**0.90**	0.89

Number of positives: SGC 141, RTV 120, LTV 93

Intersection B(1) (1 hour)

	TP			FP			FN			Precision			Recall		
	SGC	RTV	LTV	SGC	RTV	LTV	SGC	RTV	LTV	SGC	RTV	LTV	SGC	RTV	LTV
RGB	**52**	237	175	99	**8**	**1**	**19**	116	35	0.34	**0.97**	**0.99**	**0.73**	0.67	0.83
T	48	276	144	26	51	3	23	77	66	0.65	0.84	0.98	0.68	0.78	0.69
RGBT	48	**301**	**200**	**24**	16	5	23	**52**	**10**	**0.67**	0.95	0.98	0.68	**0.85**	**0.95**

Number of positives: SGC 71, RTV 353, LTV 210

Intersection C(1) (1 hour)

	TP			FP			FN			Precision			Recall		
	SGC	RTV	LTV	SGC	RTV	LTV	SGC	RTV	LTV	SGC	RTV	LTV	SGC	RTV	LTV
RGB	**54**	**109**	**94**	6	**7**	**3**	**20**	**7**	**5**	0.90	**0.94**	**0.97**	**0.73**	**0.94**	**0.95**
T	49	106	80	6	14	**3**	25	10	19	0.89	0.88	0.96	0.66	0.91	0.81
RGBT	52	108	93	**2**	**7**	4	22	8	6	**0.96**	**0.94**	0.96	0.70	0.93	0.94

Number of positives: SGC 74, RTV 116, LTV 99

Table H.3: Detection performance of the RTV, LTV, and SGC detectors evaluated at four different video sequences. A detection is marked as a true positive if it is within $\pm$ 2 seconds of the ground truth.

[6] C. Hydén, "The development of a method for traffic safety evaluation: The swedish traffic conflicts technique," *BULLETIN LUND INSTITUTE OF TECHNOLOGY, DEPARTMENT*, no. 70, 1987.

[7] S. Jackson, L. F. Miranda-Moreno, P. St-Aubin, and N. Saunier, "Flexible, mobile video camera system and open source video analysis software for road safety and behavioral analysis," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2365, no. 1, pp. 90–98, 2013.

[8] A. Laureshyn, *Application of Automated Video Analysis to Road User Behaviour*. Lund University, 2010.

[9] T. K. O. Madsen, C. Bahnsen, H. Lahrmann, and T. B. Moeslund, "Automatic detection of conflicts at signalized intersections," in *Transportation Research Board 93rd Annual Meeting*.

[10] N. Saunier, H. Ardö, J.-P. Jodoin, A. Laureshyn, M. Nilsson, Å. Svensson, L. Miranda-Moreno, G.-A. Bilodeau, and K. Åström, "A public video dataset for road transportation applications," 2013.

[11] N. Saunier and T. Sayed, "Automated analysis of road safety with video data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2019, no. 1, pp. 57–64, 2007.

[12] N. Saunier, T. Sayed, and K. Ismail, "Large-scale automated analysis of vehicle interactions and collisions," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2147, no. 1, pp. 42–50, 2010.

[13] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 7, pp. 1442–1468, 2014.

[14] Å. Svensson and C. Hydén, "Estimating the severity of safety related behaviour," *Accident Analysis & Prevention*, vol. 38, no. 2, pp. 379–385, 2006.

[15] A. P. Tarko, "Use of crash surrogates and exceedance statistics to estimate road safety," *Accident Analysis & Prevention*, vol. 45, pp. 230–240, 2012.

Paper I

Detecting Road Users at Intersections Through Changing Weather Using RGB-Thermal Video

Chris Bahnsen and Thomas B. Moeslund

The paper has been published in the
Proceedings of the International Symposium on Visual Computing, pp. 741-751, 2015.

The layout has been revised.

Abstract

This paper compares the performance of a watch-dog system that detects road user actions in urban intersections to a KLT-based tracking system used in traffic surveillance. The two approaches are evaluated on 16 hours of video data captured by RGB and thermal cameras under challenging light and weather conditions. On this dataset, the detection performance of right turning vehicles, left turning vehicles, and straight going cyclists are evaluated. Results from both systems show good performance when detecting turning vehicles with a precision of 0.90 and above depending on environmental conditions. The detection performance of cyclists shows that further work on both systems is needed in order to obtain acceptable recall rates.

1 Introduction

Road safety is a subject of high interest amongst governments and the research community. In the European Union, for instance, it is the goal of the European Commission to halve the number of road deaths by 2020 [7]. One of the ways to increase the passive safety of a road is to improve the layout of the road based on historical data of traffic events such as police and hospital records. However, not all accidents are reported and when conflict data is sparse, it might be difficult to assess the safety of a particular road. Video data, on the other hand, allows the generation of detailed information about the road users such as trajectories, speed profiles, and road user types.

However, when conflict data is sparse, one must look through thousands of hours of video to extract events of interest. Another approach is *surrogate safety analysis* which studies potential conflicts as a surrogate for real conflicts. The foundation of surrogate safety analysis is the existence of a continuous relationship between the severity of the interactions and the volume which forms a so-called conflict pyramid [10]. The fatal injuries reside on the top of the pyramid and resembles a low volume of the traffic data while normal traffic with no conflicts make up the majority of the traffic and resides in the lower part of the pyramid. Surrogate safety analysis builds on the claim that the analysis of near-conflicts gives a surrogate measure of the number of fatal interactions between road users [22].

The long-time goal of this work is to obtain a surrogate measure of the safety of bicyclists, pedestrians, and other vulnerable road users at urban intersections. In order to do this, one must reliably detect and track all road users in intersections by automatic analysis of video data. This paper presents a thorough evaluation of the block-based road user action detection technique presented in [2] on 16 hours of thermal and RGB video in three urban intersections. The data set includes a variety of sub-optimal conditions for visual algorithms, including rain, hard shadows, reflections, low lighting,

and occlusion. We compare the mentioned approach to a feature-based Kanade-Lucas-Tomasi (KLT) tracker for traffic analysis presented by Saunier et al. [20] which is made available through the open-source *trafficintelligence* project [19]. To the best of our knowledge, this is the first cross-evaluation of tracking algorithms for infrastructure-side monitoring on real-world, non-optimal thermal-visible video data.

The following section of this paper contains an overview of related work in infrastructure-side traffic surveillance. Section 3 outlines the main methods used for the block-based road user action detection system used in [2] and Section 4 explains the KLT-based feature tracker used for comparison. In Section 5, the thermal-visible dataset is described, including context and weather information of the data. Section 6 contains the experimental results of using the mentioned algorithms for cyclist and vehicle detection and Section 7 concludes the work.

2 Related Work

Traditional computer-vision based methods on traffic surveillance concerns the monitoring of motorized vehicles at highways [4]. In highways, the detection and tracking of vehicles is easier because they are usually well separated, run in separate lanes, and follow certain routes. A comprehensive survey of traffic surveillance in highway applications is found in [12].

In the past decade, researchers have explored the more complex task of monitoring vehicles at urban areas which includes monitoring of intersections [23], [20] and pedestrians and two-wheelers such as mopeds and cyclists [15], [16], [24]. Monitoring urban traffic is challenging due to the density of the traffic, variable types of road users, and lower camera orientations which aggravates occlusion. Due to the vast amount of challenges, the field of traffic analysis in computer vision is very diverse and includes a broad range of approaches. In their extensive review of urban traffic analysis with computer vision, Buch et al. divides the field into two main approaches; top-down and bottom-up surveillance systems which eventually are combined with a tracking system [5].

The foundation of *top-down surveillance* is the segmentation of the foreground which is accomplished by using a variety of classic techniques, including frame differencing, background averaging, Kalman filtering, and the Gaussian mixture model (GMM). Foreground segmentation is followed by grouping and vehicle classification which includes region- and contour-based features, and advanced machine learning. Examples of top-down approaches are found in [1], [23], and [11] where the authors use a background model to detect vehicles and [14] which is based on frame differencing.

In *bottom-up surveillance*, the foreground segmentation is replaced by patch

detectors and classifiers. Examples in traffic surveillance include Hessian corners [20], SIFT [25], and boosting [13].

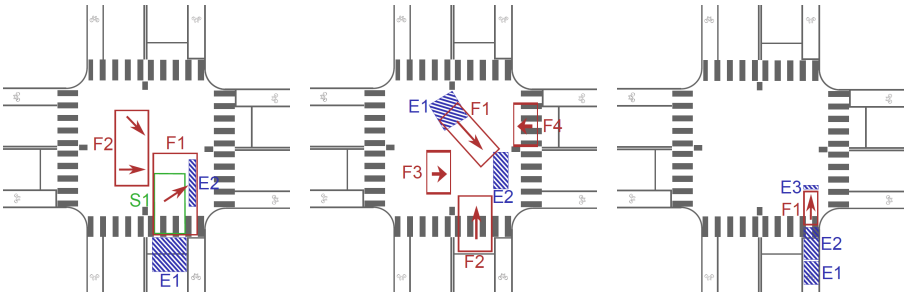
Tracking is used to connect observations of road users in consecutive frames into spatio-temporal trajectories. The classic Kalman filter is used in a variety of applications, including [15]. Trackers based on the Kalman filter assumes a Gaussian process and measurement noise, which is not fulfilled in the general case of tracking urban traffic. The Particle Filter removes these assumptions at the cost of computational simplicity and is used for tracking motorcycles in [18]. Saunier and Sayed use the KLT tracker to track keypoints of vehicles in intersections [20]. Tracked features are grouped over time according to the spatial distance of the tracks. The work of Saunier and Sayed has been used in [21] to predict collision amongst vehicles in intersections and extended in [24] to include the classification of road users, including pedestrians and bicycles.

3 Watch-Dog Detection of Road User Actions

Detection, tracking, and classification of urban traffic in unconstrained scenarios pose a substantial challenge to computer vision algorithms. Existing methods are typically evaluated either at short intervals or under ideal conditions. An automated system which is able to accomplish these tasks in unconstrained scenarios does currently not exist [5]. On the other hand, manual monitoring of vast amounts of video is an expensive and tedious task which indeed does not scale well to analysis of complex transport networks. In the recent work of [2], the authors take steps to close the gap between automated and manual analysis by introducing a semi-automated watch-dog system, whose aim is to reduce the amount of video data for inspection by the traffic analysts. This semi-automated system is specialized for the detection of interactions between Right Turning Vehicles (RTV), Left Turning Vehicles (LTV), and Straight Going Cyclists (SGC) at intersections. The goal of the watch-dog is not to perform perfect tracking of road users but to obtain a reasonable data reduction.

The watch-dog system contains a cascade of two fundamental detector types that registers either presence or movement in a region of interest (ROI). Each fundamental detector is laid out in a predefined ROI where the road users of interest may be observed. In the watch-dog system, a detector is either triggered or non-triggered, and it is a combination of this binary logic that lays out the RTV, LTV, and SGC detectors.

The *presence detector* uses a background subtraction technique based on reference images. The reference images are compared to the current frame by computing the Canny edges [6] in the ROI of the frame. If the difference of the two edge images is greater than a specified threshold, the detector is



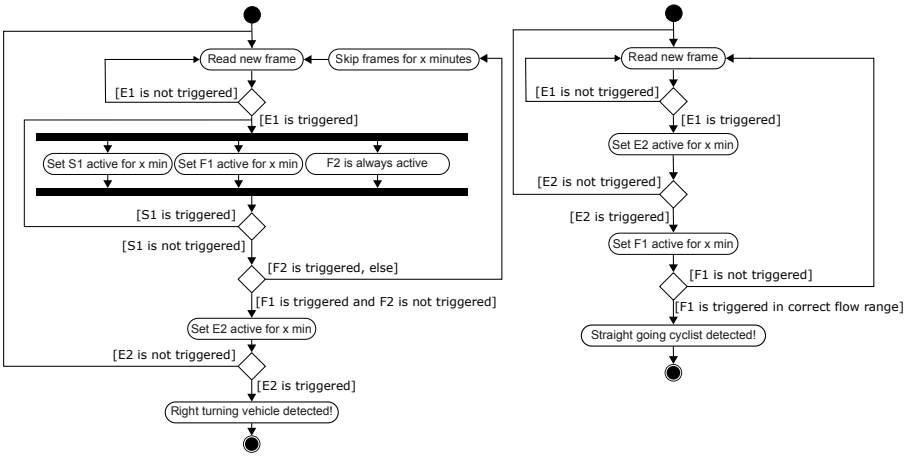
(a) RTV **(b)** LTV **(c)** SGC

Fig. I.1: Prototype layouts for detecting road user actions. The arrows of the F detectors indicates the direction of movement for which the detector is configured

marked as triggered. If the difference of the edge images is below 80 % of the threshold, and the background has not been updated for τ consecutive frames, the background image is updated with the current image.

The *movement detector* estimates the movement in a certain ROI of the intersection by computing the dense optical flow [8] between two frames. Only the flow vectors within a desired direction of movement and above a certain magnitude are kept. If the number of remaining flow vectors surpass a threshold, the detector is triggered. A detailed description of the movement and presence detectors is found in [2].

The movement and presence detectors are overlaid on specific parts of the intersection and several detectors are chained to detect RTV, LTV, or SGC. For instance, if we want to detect RTV, we know that vehicles of interest must enter the intersection, perform a right turn in a designated area, and eventually exit the intersection. In the watch-dog framework, this translates to three sequential detections; detecting presence, detecting movement, and detecting presence. Prototype layouts of the RTV, LTV and SGC detectors are shown in Figure I.1. The presence detector is abbreviated E (edge), the movement detector F (flow), and a new S (stationary) detector is introduced which detects if something is present, but not moving. The stationary detector is a combination of the presence and movement detectors configured on the same ROI. Activity diagrams which describe the sequential logic of the RTV and SGC detectors, are shown in Figure I.2. The LTV and RTV detectors contain one or more modules (F2, F3, F4) which are used to prevent the detection of road users from other directions.



(a) RTV detector **(b)** SGC detector

Fig. I.2: Activity diagrams of RTV and SGC watch-dog detectors. The LTV detector is similar to the RTV detector. In the LTV detector, F3 and F4 shares the behaviour of F2 in the RTV detector

3.1 Fusing Modalities

The watch-dog operates on RGB, thermal, and combined RGBT video data. In RGBT mode, the fundamental detectors of the watch-dog run in parallel on the synchronized RGB and thermal video data. The fundamental detectors output a confidence value between 0 and 1 and a value ≥ 0.5 indicates that the detector is triggered. If the averaged confidence value of the RGB and thermal detectors is above 0.5, the multi-modal detector is triggered.

4 Feature-Based Tracking of Road Users

We use the feature-based tracker of Saunier et al. [20] to compare the results of the watch-dog and assess the method in less-than-ideal weather conditions. The feature-based tracking algorithm is an extension of the method by Beymer et al. [4] which was used to track vehicles at highways. The cornerstone of the tracking algorithm is the KLT-tracker [3]. The extracted features are tracked over time and in order to mimic the physical constraints of the movement of the road users, the features are kept only if the spatio-temporal displacement is small and the averaged motion of the features is smooth. The displacement and motion constraints also entail that only objects in motion are tracked, i.e. if road users stop at any point in the intersection, the tracking of the road user is suspended and will be initiated under a different ID once the road user starts moving.

The features from the tracking stage are grouped in a subsequent, offline step. The grouping algorithm operates in world coordinates and groups feature tracks with similar motion. A feature is grouped to nearby features if the distance is below the maximum distance threshold, $D_{\text{connection}}$. A new feature is easily grouped to several feature groups through this connection stage. However, new features are only added to existing groups if the feature and the group are similar for a minimum number of frames.

For each frame, it is checked if the available pairs of connected features are still belonging together. The distance $d_{i,j}$ between the connected features is computed and it is checked if the relative motion between the two feature tracks is within a segmentation threshold, $D_{\text{segmentation}}$. If their relative motion is above $d_{\text{segmentation}}$, the features are disconnected.

The $D_{\text{connection}}$ and $D_{\text{segmentation}}$ thresholds are tuned to obtain a balance between overgrouping and oversegmentation. If the thresholds are set too low, road users will be oversegmented, i.e. a single road user will be represented as multiple tracks. If the thresholds are set too high, adjacent road users are detected as one. Finding the right thresholds is a challenge, however, and one has to choose the road user type for which the thresholds should be optimized. For instance, the algorithm might correctly detect car-sized objects while smaller road user types, such as cyclists, are prone to overgrouping and larger objects, such as lorries, are oversegmented.

4.1 Fusing modalities

Although the feature-based tracker of [20] is built to operate on RGB video, it also translates well to video in the thermal domain. As described in the following section, objects in thermal video generally contain less information than their RGB representation. This is taken into account by adjusting the KLT-tracking parameters and allowing the formation of trajectory groups with fewer trajectories. Additional grouping parameters need not be changed because the grouping is performed in world coordinates in both domains. In RGBT mode, features are extracted separately in the RGB and thermal modality and mapped to a common world coordinate system. Grouping is then performed on the combined RGB and thermal trajectories to produce one single output.

In order to compare the performance of the watch-dog and feature-based tracking, we need to obtain measures of RTV, LTV, and SGC from the latter. Because the entry and exit points of these road users are well-defined in intersections, we define entry and exit masks for each road user action type. We thus define a RTV from an object trajectory if the trajectory passes through entry and exit masks defined for the intersection in question.

5 Thermal-Visible Intersection Data Set

In order to track road users in a diverse range of weather and environmental conditions, the road users themselves must be visible to the algorithms. This is difficult during the night if artificial illumination is sparse. Vehicles might be detected by their headlights - but what about pedestrians and bicyclists?

Thermal cameras are independent of the availability of visible light and only depends on the emitted radiation from objects. Thus, thermal cameras allows to see objects through the night, as long as the temperature of the objects is different from the temperature of the surroundings. Contrary to RGB cameras, thermal cameras are not susceptible to shadows. However, features are sparse in thermal images, and it might thus be more difficult to determine identities or distinguish between objects. An extensive review of thermal cameras in computer vision is found in [9]. When combined, RGB and thermal cameras extend the visibility of the road users and improves the robustness of traffic surveillance algorithms.

The proposed data set is an extension of the data set used in [2]. We extend the original four hours of video data to 16 hours and include a broader variety of weather and lighting conditions such as rain, wind, twilight, overcast, and full daylight. Further details regarding the contextual information of the data set is found in Table I.1. Samples of each of the three locations are shown in Figure I.3. For each of the three locations, RTV, LTV, and SGC have been manually annotated and assigned a time stamp whenever the desired road user type enters the area of the intersection corresponding to the E2 or E3 module of the watch-dog detectors shown in Figure H.2.

6 Experimental Results

The watch-dog and the feature-based tracker are applied on the 16 hours of video described in Table I.1. In order to mitigate the oversegmentation of vehicles caused by the feature-based tracker, duplicate tracks are filtered in a post-processing step. Only one object trajectory per second is allowed to pass through what corresponds to the E2 and E3 module of the watch-dog detectors and other tracks within the second are filtered out. Because oversegmentation is not a problem amongst cyclists, the 1 second filter is only applied to the detection of RTV and LTV. For detection of SGC, the threshold is relaxed to 0.3 seconds. A detection is marked as a true positive if is within $\pm$ 2 seconds of the ground truth.

The results of the test are shown in Table I.2. It is seen that the watch-dog based detection of RTV and LTV generally is robust with precision and recall rates above 0.90 in several sequences. The RGBT mode of the watch-dog is shown to be stable whenever each individual modality gives acceptable results.

Table I.1: Environmental conditions of the proposed data set. The sequences are distributed over three days for each intersection

Location	Seq.	Time of day	Weather	Temp.	Lighting
A	1	07:00 - 08:00	Partly cloudy	12 °C	Full daylight
A	2	07:00 - 08:00	Light rain	17 °C	Overcast
A	3	07:00 - 08:00	Mostly Cloudy	15 °C	Overcast
A	4	12:00 - 13:00	Clear	19 °C	Full daylight
A	5	15:00 - 16:00	Light rain	19 °C	Overcast
A	6	16:00 - 17:00	Mostly Cloudy	19 °C	Full daylight
B	1	06:00 - 07:00	Rain	12 °C	Twilight
B	2	07:00 - 08:00	Rain	12 °C	Overcast
B	3	07:00 - 08:00	Shallow Fog, Partly Cloudy	6 °C	Overcast
B	4	12:00 - 13:00	Mostly Cloudy	13 °C	Full daylight
B	5	16:00 - 17:00	Partly Cloudy	17 °C	Full daylight
C	1	07:00 - 08:00	Light Rain Showers	13 °C	Overcast
C	2	07:00 - 08:00	Mostly Cloudy	13 °C	Overcast
C	3	12:00 - 13:00	Mostly Cloudy	16 °C	Overcast
C	4	16:00 - 17:00	Light Rain Showers	14 °C	Overcast
C	5	21:00 - 22:00	Rain Showers	12 °C	Deep twilight

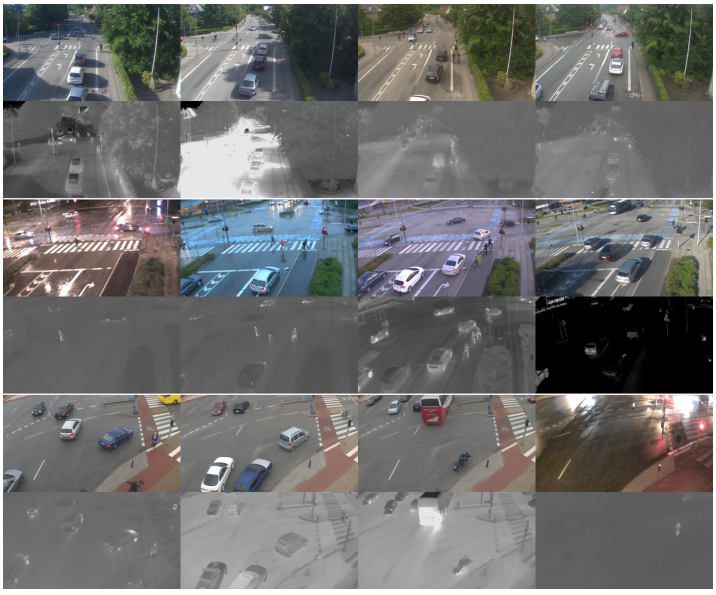


Fig. I.3: Sample images of intersection A (top), B (middle), and C (bottom). The appearance of the intersections in both modalities vary greatly due to changing environmental conditions

Whenever the watch-dog fail to detect road users in a single modality, the RGBT results will suffer accordingly. Whenever results are stable, the RGBT mode performs best or trails behind the best performing modality by few percentage points. The results of the feature-based tracker shows remarkable precision with few or none false positives. Recall of RTV and LTV shows to be comparable or slightly better than the watch-dog performance. However, the detection of SGC by either the watch-dog or feature-based approaches shows considerable room for improvement. The smaller size and the irregular motion of the SGC are still challenges that need to be solved.

7 Conclusion

This work evaluates the detection performance of left and right turning vehicles and straight going cyclists at urban intersections. Two detection approaches are evaluated at 16 hours of RGB and thermal video data featuring challenging weather and light levels. The first approach, the watch-dog, detects road user actions by using a chained set of basic detectors and spatial constraints of the intersection. The second approach, the feature-based detector, uses a KLT-tracker and additional grouping to track moving objects in the intersection. Both approaches show promising results when detecting vehicles while the detection of cyclists shows room for further improvement. The use of RGB and thermal modalities generally results in more stable performance for both detection approaches. However, more sophisticated weighting of modalities is needed to filter out false negatives whenever a detection algorithm breaks down in one modality.

Future work includes more persistent tracking of road users at all speeds in the intersection and further road user classification. Once full trajectories are found, trajectory classification techniques will be investigated to gather more detailed information of the road user actions [17].

Acknowledgements

The authors thank Tanja Kidmann Osmann Madsen for acquiring the data as well as assistance on the ground truth. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 635895. This publication reflects only the author's view. The European Commission is not responsible for any use that may be made of the information it contains.

Table I.2: RTV, LTV, and SGC detection performance of the watch-dog and feature-based trackers. The number of manually annotated SGC, RTV, and LTV road user actions is marked in italics in the right side of the table

Seq.		Block-based Watch-dog						Feature-based tracker						GT
		Precision			Recall			Precision			Recall			
		SGC	RTV	LTV	SGC	RTV	LTV	SGC	RTV	LTV	SGC	RTV	LTV	
A1	RGB	0.20	0.52	0.93	**0.71**	0.50	0.84	0.48	0.72	**0.91**	0.27	**0.80**	0.62	*146*
	T	0.77	**0.98**	0.94	0.70	0.75	0.65	**0.96**	**0.96**	0.77	**0.32**	0.79	**0.97**	*122*
	RGBT	**0.80**	0.97	**0.96**	**0.71**	**0.80**	**0.92**	0.57	0.74	0.86	0.24	0.75	0.92	*77*
A2	RGB	0.29	0.98	**0.99**	**0.64**	0.78	**0.99**	0.95	0.96	0.88	0.43	0.81	0.96	*131*
	T	0.57	**0.99**	0.91	0.56	0.61	0.55	**1.00**	**0.99**	0.93	0.19	0.69	0.96	*130*
	RGBT	**0.76**	0.97	0.97	0.60	**0.84**	0.89	0.95	0.95	**0.94**	**0.44**	**0.82**	**0.97**	*74*
A3	RGB	0.68	0.98	0.91	0.65	**0.90**	**0.97**	0.93	**0.99**	**0.87**	0.47	**0.88**	0.87	*141*
	T	0.85	0.98	0.94	0.65	0.85	0.16	**0.96**	**0.99**	0.83	0.33	0.70	**0.95**	*120*
	RGBT	**0.90**	**1.00**	**0.97**	**0.69**	**0.90**	0.89	0.95	0.97	0.85	**0.52**	0.87	0.94	*93*
A4	RGB	0.11	**1.00**	**0.91**	**0.90**	0.50	0.87	0.75	0.98	**0.93**	0.10	**0.88**	0.68	*29*
	T	0.73	0.97	0.90	0.55	**0.83**	0.87	**1.00**	**1.00**	0.84	0.10	0.77	**0.99**	*101*
	RGBT	**0.77**	0.99	0.90	0.79	0.79	**0.96**	0.83	0.99	0.91	**0.17**	0.87	**0.99**	*82*
A5	RGB	0.54	**0.98**	0.95	**0.70**	0.78	**0.91**	0.95	**0.98**	**0.93**	**0.41**	**0.88**	0.85	*44*
	T	0.63	0.94	0.90	**0.70**	0.78	0.27	**1.00**	0.96	0.91	0.07	0.51	**0.90**	*156*
	RGBT	**0.66**	0.97	**0.99**	**0.70**	**0.91**	0.73	0.92	0.96	0.91	0.27	0.77	0.88	*139*
A6	RGB	**1.00**	**0.96**	**0.98**	0.08	**0.80**	0.88	**1.00**	0.97	**0.91**	**0.18**	**0.90**	0.86	*39*
	T	0.27	0.73	0.96	**0.90**	0.27	0.88	**1.00**	**0.98**	0.89	0.15	0.82	**0.99**	*137*
	RGBT	**1.00**	0.95	0.97	0.00	0.42	**0.96**	**1.00**	**0.98**	0.89	0.15	0.82	**0.99**	*155*
B1	RGB	0.19	**0.97**	**0.98**	0.82	0.55	0.48	**1.00**	**0.94**	0.92	0.71	**0.85**	**1.00**	*28*
	T	**0.78**	0.91	0.97	**0.89**	0.71	0.69	**1.00**	0.93	**0.95**	0.64	0.67	0.70	*195*
	RGBT	0.63	0.94	0.96	0.86	**0.78**	**0.82**	0.96	0.92	0.62	**0.79**	0.81	0.65	*89*
B2	RGB	0.34	**0.97**	**0.99**	**0.73**	0.67	0.83	**1.00**	**0.97**	**0.93**	**0.72**	**0.88**	**0.97**	*71*
	T	0.65	0.84	0.98	0.68	0.78	0.69	0.98	**0.97**	0.92	0.61	0.75	0.91	*353*
	RGBT	**0.67**	0.95	0.98	0.68	**0.85**	**0.95**	**1.00**	**0.97**	0.83	0.69	0.83	0.79	*210*
B3	RGB	0.43	**0.99**	0.92	**0.65**	0.89	0.93	**1.00**	**1.00**	0.90	0.63	0.84	**0.98**	*92*
	T	0.19	0.94	0.92	0.64	0.80	0.83	0.98	0.99	0.71	**0.68**	**0.87**	0.81	*377*
	RGBT	**0.49**	**0.99**	**0.93**	0.63	**0.90**	**0.99**	**1.00**	**1.00**	**0.91**	0.61	0.81	0.88	*177*
B4	RGB	**0.08**	0.98	0.96	**0.82**	**0.67**	**0.89**	**1.00**	0.96	0.93	**0.71**	**0.81**	**0.99**	*28*
	T	0.03	**1.00**	**1.00**	0.79	0.00	0.02	**1.00**	**0.99**	0.86	0.64	0.78	**0.99**	*205*
	RGBT	0.04	**1.00**	**1.00**	0.79	0.00	0.05	**1.00**	0.98	**0.95**	**0.71**	0.75	**0.99**	*87*
B5	RGB	0.09	**0.99**	0.97	**0.83**	0.88	**0.94**	0.83	**0.99**	0.95	0.60	0.83	**0.99**	*48*
	T	0.26	**0.99**	**1.00**	0.60	0.91	0.87	**0.95**	0.96	0.87	**0.77**	**0.89**	0.91	*347*
	RGBT	0.07	**0.99**	0.99	0.81	**0.93**	**0.94**	0.86	**0.99**	**0.98**	0.63	0.80	0.95	*102*
C1	RGB	0.90	**0.94**	**0.97**	**0.73**	**0.94**	**0.95**	**1.00**	0.96	0.96	0.51	0.94	0.69	*74*
	T	0.89	0.88	0.96	0.66	0.91	0.81	**1.00**	0.93	0.93	**0.54**	**0.97**	**0.96**	*116*
	RGBT	**0.96**	**0.94**	0.96	0.70	0.93	0.94	**1.00**	**0.97**	**0.97**	0.53	**0.97**	0.70	*99*
C2	RGB	0.80	0.97	0.94	**0.75**	0.94	0.81	**0.98**	0.97	0.98	**0.52**	0.97	0.72	*88*
	T	0.93	0.42	**0.97**	0.74	0.96	0.88	**0.98**	0.94	0.94	0.51	**1.00**	**0.96**	*120*
	RGBT	**0.94**	**0.98**	0.95	**0.75**	**0.97**	**0.92**	**0.98**	**0.99**	**0.99**	**0.52**	0.97	0.73	*113*
C3	RGB	**0.79**	**0.99**	**0.94**	0.92	**0.94**	**0.94**	0.91	**1.00**	0.98	**0.83**	0.97	0.82	*12*
	T	0.03	0.55	0.81	**1.00**	0.46	0.48	**1.00**	0.98	0.98	**0.83**	**1.00**	**0.91**	*128*
	RGBT	0.14	0.98	**0.94**	0.75	0.48	0.61	0.91	**1.00**	**0.99**	**0.83**	0.99	0.77	*109*
C4	RGB	**0.84**	**0.98**	**0.99**	**0.79**	0.35	**0.93**	**1.00**	**1.00**	**0.99**	0.56	0.96	0.82	*34*
	T	0.31	0.93	0.88	**0.79**	**0.74**	0.81	**1.00**	0.97	0.98	0.65	**0.98**	**0.93**	*155*
	RGBT	0.33	0.96	**0.99**	0.76	0.61	0.81	**1.00**	0.99	**0.99**	**0.62**	0.97	0.80	*103*
C5	RGB	0.01	0.53	**1.00**	**0.67**	0.24	0.16	**1.00**	0.97	**1.00**	**0.67**	0.94	0.84	*3*
	T	**0.67**	0.94	**1.00**	**0.67**	0.88	0.68	0.67	0.92	**1.00**	**0.67**	**1.00**	**0.95**	*33*
	RGBT	**0.67**	**1.00**	0.93	**0.67**	**1.00**	**0.74**	**1.00**	**1.00**	**1.00**	**0.67**	**1.00**	**0.95**	*19*

References

[1] Ö. Aköz and M. E. Karsligil, "Traffic event classification at intersections based on the severity of abnormality," *Machine vision and applications*, pp. 1–20, 2014.

[2] C. Bahnsen and T. B. Moeslund, "Detecting road user actions in traffic intersections using rgb and thermal video," in *Advanced Video and Signal Based Surveillance, 2015. AVSS 2015. IEEE Conference on*. IEEE, 2015.

[3] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[4] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, "A real-time computer vision system for measuring traffic parameters," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. IEEE, 1997, pp. 495–501.

[5] N. Buch, S. Velastin, J. Orwell *et al.*, "A review of computer vision techniques for the analysis of urban traffic," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 3, pp. 920–939, 2011.

[6] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 679–698, 1986.

[7] European Commission, "White paper roadmap to a single european transport area towards a competitive and resource efficient transport system," *COM (2011)*, vol. 144, 2011.

[8] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image Analysis*. Springer, 2003, pp. 363–370.

[9] R. Gade and T. B. Moeslund, "Thermal cameras and applications: A survey," *Machine vision and applications*, vol. 25, no. 1, pp. 245–262, 2014.

[10] C. Hydén, "The development of a method for traffic safety evaluation: The swedish traffic conflicts technique," *BULLETIN LUND INSTITUTE OF TECHNOLOGY, DEPARTMENT*, no. 70, 1987.

[11] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, "Traffic monitoring and accident detection at intersections," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 1, no. 2, pp. 108–118, 2000.

[12] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis, "A survey of video processing techniques for traffic applications," *Image and vision computing*, vol. 21, no. 4, pp. 359–381, 2003.

[13] A. Khammari, F. Nashashibi, Y. Abramson, and C. Laurgeau, "Vehicle detection combining gradient analysis and adaboost classification," in *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*. IEEE, 2005, pp. 66–71.

[14] Y.-K. Ki and D.-Y. Lee, "A traffic accident recording and reporting model at intersections," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 8, no. 2, pp. 188–194, 2007.

[15] B. Maurin, O. Masoud, and N. P. Papanikolopoulos, "Tracking all traffic: computer vision algorithms for monitoring vehicles, individuals, and crowds," *Robotics & Automation Magazine, IEEE*, vol. 12, no. 1, pp. 29–36, 2005.

[16] S. Messelodi, C. M. Modena, and M. Zanin, "A computer vision system for the detection and classification of vehicles at urban road intersections," *Pattern analysis and applications*, vol. 8, no. 1-2, pp. 17–31, 2005.

[17] B. T. Morris and M. M. Trivedi, "Learning, modeling, and classification of vehicle track patterns from live video," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 9, no. 3, pp. 425–437, 2008.

[18] P.-V. Nguyen and H.-B. Le, "A multi-modal particle filter based motorcycle tracking system," in *PRICAI 2008: Trends in Artificial Intelligence*. Springer, 2008, pp. 819–828.

[19] N. Saunier, ""trafficintelligence"," https://bitbucket.org/Nicolas/trafficintelligence, 2015, accessed: 2015-07-29.

[20] N. Saunier and T. Sayed, "A feature-based tracking algorithm for vehicles in intersections," in *Computer and Robot Vision, 2006. The 3rd Canadian Conference on*. IEEE, 2006, pp. 59–59.

[21] N. Saunier, T. Sayed, and K. Ismail, "Large-scale automated analysis of vehicle interactions and collisions," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2147, no. 1, pp. 42–50, 2010.

[22] Å. Svensson and C. Hydén, "Estimating the severity of safety related behaviour," *Accident Analysis & Prevention*, vol. 38, no. 2, pp. 379–385, 2006.

[23] H. Veeraraghavan, O. Masoud, and N. P. Papanikolopoulos, "Computer vision algorithms for intersection monitoring," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 4, no. 2, pp. 78–89, 2003.

[24] S. Zangenehpour, L. F. Miranda-Moreno, and N. Saunier, "Automated classification in traffic video at intersections with heavy pedestrian and bicycle traffic," in *Transportation Research Board 93rd Annual Meeting*, no. 14-4337, 2014.

[25] W. Zhang, B. Yu, G. J. Zelinsky, and D. Samaras, "Object class recognition using multiple layer boosting with heterogeneous features," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 323–330.

Paper J

Automatic Detection Of Conflicts At Signalized Intersections

Tanja K. O. Madsen, Chris Bahnsen, Harry Lahrmann, and Thomas B. Moeslund

This abstract has been presented at the
Transportation Research Board 93rd Annual Meeting: Workshop on Comparison of Surrogate Measures of Safety Extracted from Video Data, 2014

The layout has been revised.

1 Background and purpose

The Swedish Conflict Technique [3] is a well-known way of assessing the safety of a particular road design. In recent years the application of automatic image analysis instead of human observers has been exam-ined, e.g. in [5].

The scope of this work is to develop a method for automatic image analysis using a double input from RGB and thermal cameras to provide time stamps of potential conflicts and traffic counts. Subsequently, an in-depth analysis of the potential conflicts is performed manually.

Studies have shown that the construction of bike paths results in a higher number of accidents with cyclists in intersections and in particular in those controlled by traffic lights [1], [4]. Through the years different designs of bike paths in signalized intersections have been established in order to improve the safety of cyclists. However, there is no clear evidence of when the different bike path solutions should be used and whether the best bike path solution differs with varying traffic volumes. In this work the assessment of the safety of cyclists is based on the number of near-collisions between cyclists and left/right turning cars. To facilitate the detection of potential near-collisions, video analysis techniques have been applied in this work. Concretely, a comprehensive case study compares five different designs of bike paths in signalized intersections.

2 The method

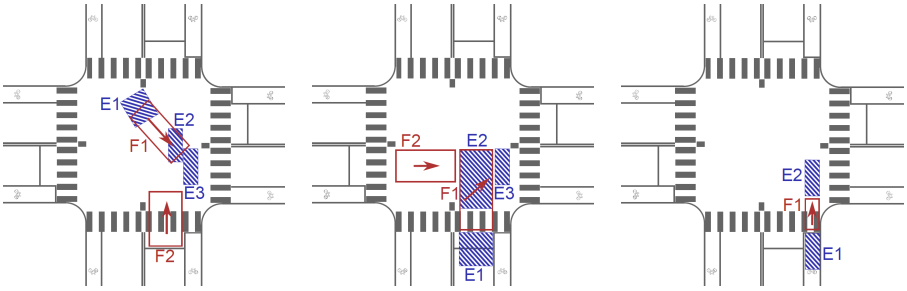
Two situations are of special interest to detect:

1. The time gap between a car and a cyclist with crossing trajectories is small
2. One or both of the road users stops near the intersection point between their trajectories in order to avoid a potential collision between the two road users

A tool using multi-modal imagery for automatic detection of interactions, and thus potential conflicts, has been developed. The tool utilizes a combination of RGB and thermal imagery. Whereas the RGB camera is able to capture a higher level of detail of the road users, the thermal camera might detect road users which would be invisible to the RGB camera due to shadows or result in false positives, see Figure J.1. A full utilization of the thermal camera requires a synchronization of the RGB and thermal recordings, which is solved in a post-processing step.

The tool measures the post-encroachment time of cars and cyclists in the conflict area, which is manually defined for each intersection. The detection of potential conflicts is split into the subtasks of detecting:

Fig. J.1: Shadows complicates the detection of a cyclist in the RGB recordings, whereas the cyclist is easily distinguished from the background in the thermal recording.



(a) Left Turning Cars **(b)** Right Turning Cars **(c)** Straight Going Cyclists

Fig. J.2: (a) Detection of left turning cars. Consists of three edge detectors and two flow detectors. (b) Detection of right turning cars. Consists of three edge detectors and two flow detectors. (c) Detection of straight going cyclists. Consists of two edge detectors and one flow detector.

- Straight going cyclists (SGC)
- Right turning cars (RTC)
- Left turning cars (LTC)

The output of these subtasks is used for the detection of potential conflicts and the computation of the time gap between the car and the cyclist. The detection of these subtasks is handled by module-based program logic that consists of two blocks; flow detectors (F) and edge detectors (E). The configuration of each subtask is seen from Figure J.2 below.

The flow detectors compute the dense optical flow of the predefined areas by using the algorithm of [2].

	Left turning car	Right turning car	Straight going cyclist
E1	On	On	On
E2	On	On	On
E3	On	On	On
F1	On	On	-
F2	Off	Off	-

Table J.1: Relationship between detection subtasks and their corresponding detector blocs. A detection subtask is triggered when the individual detector blocs (E1-E3, F1-F2) are triggered in the order shown above.

The flow vectors of the algorithm are filtered such that only vectors within a specific angle range are counted. The flow detector is triggered whenever the number of vectors inside this range exceeds a predefined threshold. The average flow vector is of each region is illustrated by a red arrow in Figure J.2. Whereas the flow detectors determine if something is moving in a certain direction, the edge detectors determine if something is present within its region. The edge detectors rely on a background subtraction algorithm that uses the edge detection of [6]. Whenever the amount of edges becomes significant, the detector is triggered. If the amount of edges is not significant, the background is updated.

In order to detect a specific road user action, the blocks must be triggered in a pre-determined order that is specific to each detection subtask. The subtask of detecting left and right turning cars is distributed upon three edge detectors (E1, E2, E3) and two flow detectors (F1, F2). In order to detect a left or right turning car, the blocks E1, F1, E2, and E3 must be triggered in succession. If any activity in F2 is detected, the before mentioned blocks are deactivated for an interval of time to prevent false positives.

The subtask of detecting straight going cyclists is accomplished using two edge detectors (E1, E2) and one flow detector (F1). The detectors E1, F1, and E2 are activated in succession. If E2 is triggered, a straight going cyclist is detected. The logic of the three detection subtasks is listed in Table J.1.

A potential conflict is detected if:

1. A cyclist enters the conflict zone less than 2.5 seconds after a car has left the zone
2. A cyclist leaves the conflict zone less than 1.0 seconds before a car enters the zone
3. A car stops near the conflict area while a cyclist is present in the conflict area

In the detection subtasks of Figure J.2a and J.2b, the conflict zone is the edge detector block E3. In the detection subtask of Figure J.2c, the conflict

zone is E2. The specific layout of the masks is dependent of the properties of the intersection. The conflict zones of the car and cyclist detectors should overlap but not necessarily be identical.

3 Application of the method

The method described is expected to be capable of detecting the presence of road users on predetermined boundaries; however, it is still under development and needs to be validated. In the validation the results from the developed software will be compared to a manual detection of the potential conflicts. Then the method will be used in the comparison of the number of conflicts in intersections with different designs of the bike path across the intersection. At the workshop we will present preliminary results for these phases of the project.

Acknowledgements

The work was funded by The Danish Road Directorate. The authors wish to thank Aliaksei Laureshyn (Lund University) for advice and contributions regarding video recordings and data analysis.

References

[1] N. Agerholm, S. Caspersen, J. C. O. Madsen, and H. Lahrmann, "Cykelstiers trafiksikkerhed: en før-efterundersøgelse af 46 nye cykelstiers sikkerhedsmæssige effekta before-after study of the safety effect of 46 new cycle tracks," *Dansk Vejtidsskrift*, vol. 83, no. 12, pp. 52–57, 2006.

[2] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Scandinavian conference on Image analysis*. Springer, 2003, pp. 363–370.

[3] C. Hydén, "The development of a method for traffic safety evaluation: The swedish traffic conflicts technique," *Bulletin Lund Institute of Technology, Department*, no. 70, 1987.

[4] S. U. Jensen, "Effekter af cykelstier og cykelbaner: Før-og-efter evaluering af trafiksikkerhed og trafikmængder ved anlæg af ensrettede cykelstier og cykelbaner i københavns kommune," *Trafitec, Lyngby*, 2006.

[5] A. Laureshyn, "Application of automated video analysis to road user behaviour," 2010.

Paper K

Road User Behaviour Analyses Based on Video Detections: Status and Best Practice Examples From the RUBA Software

Charlotte Tønning, Tanja K. O. Madsen, Chris H. Bahnsen, Thomas, B. Moeslund, Niels Agerholm, and Harry Lahrmann

The paper has been published in the
Proceedings of the 24th ITS World Congress, pp. 1-10, 2017.

The layout has been revised.

Abstract

To a large extent, traffic safety improvements rely on reliable and full-covering accident registration. This is difficult to obtain in practice. Hence, surrogate measures as traffic conflict studies can contribute with more information. To make these studies more efficient, a software called RUBA has been developed. It works as a watchdog – if a passing road user affects defined part(s) of the video frame, RUBA records the time of the activity. It operates with three type of detectors (defined parts of the video frame): 1) if a road user passes the detector independent of the direction, 2) if a road user passes the area in one pre-adjusted specific direction and 3) if a road user is standing still in the detector area. Also, RUBA can be adjusted so it registers massive entities (e.g. cars) while less massive ones (e.g. cyclists) are not registered. The software has been used for various analyses of traffic behaviour: traffic counts with and without removal of different modes of transportation, traffic conflicts, traffic behaviour for specific traffic flows and modes and comparisons of speeds in rebuilt road areas. While there is still space for improvement regarding data treatment speed and user-friendliness, it is the conclusion that, at present, the RUBA software assists a number of traffic behaviour studies more efficiently and reliably than what is obtainable by human observers.

1 Background for the software development

Traffic accidents are one of the main killers in the societies. More than 1.25 million fatalities and 50 million injured are registered each year [15]. Most countries in the industrial world have experienced significant reductions in the number of fatalities since the 1970s (2). Also, a specific marked reduction has been recognised since the initiation of the financial crisis in 2007-8 [13]. However, as the crisis fades, the number of fatalities has started to rise again [2, 13], and the societies are still far from the ideal situation regarding traffic safety as described most thoroughly by the Swedish Vision 0 [14].

As focus on traffic safety increased, it also became clear that not all traffic safety problems could be identified and quantified proper from traditional traffic accident registrations [17]. This is partly because of the skewness of registration depending of accident and road user type and the general dark figures in traffic accident registration [3, 17], but also due to the limited information available in traditional traffic accident data. Therefore, surrogate measures might show a truer pattern than traditional accident data. One of the most well-reputed surrogate methods is the traffic conflict study (TCS) as thoroughly described by Hydén [6] and elaborated further on in many cases, see e.g. [11, 12, 17].

A TCS is normally made for individual locations, often intersecting ones, and the basic idea is to register any activities where absence of an avoidance activity would have resulted in an accident. This is termed 'conflict'. The time

between the avoidance activity and the accident if no avoidance was made defines if the conflict is serious or not [6,17].

Originally, the TCS registration was made by reporters, i.e. persons who monitored the traffic activities manually. The work load was high as it required one person per traffic flow to study [9]. Later, video registration took over. Cameras placed with overview on relevant parts of the location recorded the traffic behaviour. Subsequently, analyses were made based on the recorded videos. However, even though the analysis work moved to an office space, it was still very time-consuming [9]. Therefore, with the upcoming video analysis tool, more of the analysis work can be made automatically or semi-automatically. One of these software systems developed for video analyses is the 'Road User Behaviour Analysis (RUBA), which will be elaborated on here. The remaining part of this paper consists of a brief introduction to some of the available on-the-shelf products, presentation of RUBA, how it works, selected case studies and a discussion on the possibilities and shortcomings with the RUBA software as it is now.

2 An overview of on-the-shelf products

Most available products for traffic analysis come as an integrated solution for both hardware and software. There is a range of products, but the ones mentioned cover the most relevant issues. PedTrax and Smart Cycle from Iteris have their own hardware and can count and measure speed bi-directionally [7]. Traffic Flow from Viscando Traffic Systems counts different road users and can detect how road users use the recorded space [18]. DataFromSky [5] and Cowi A/S [4] use drone recordings, and can detect speed of individual vehicles and provide trajectories for the beneficiary. A few products and initiatives are available that enable end-users to analyse video recordings on their own computers, i.e. platform independent. The Traffic Intelligence project [8] allows for tracking and classification of road users from video. Recently, the functionality has been extended by the tvaLib library [16] allowing for further analysis and visualisation of the tracking results. Both of the two last-mentioned projects are primarily utilized from the command line and are thus not accessible for most end-users.

3 What is RUBA?

RUBA is a computer-based video analysis tool for Windows, Linux and MacOS. The analysis is applied to the recorded video files and is thus independent of the hardware used for the video acquisition. RUBA is developed in collaboration between the Division of Transportation Engineering and Visual Analysis of People Laboratory at Aalborg University as a part of the ongoing H2020

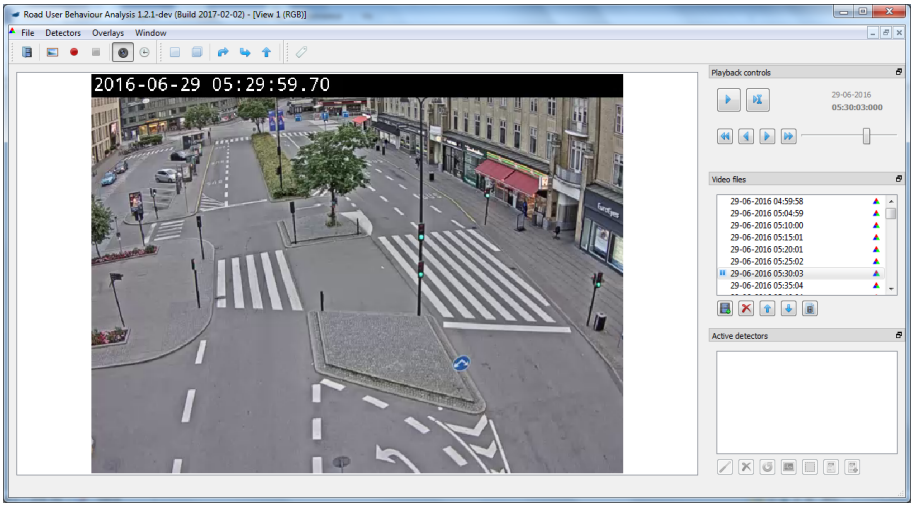


Fig. K.1: The user interface of RUBA.

project InDeV [1]. The program can be used to analyse videos recorded at a specific location where traffic-related problems need to be studied. RUBA works as a watchdog, which means that RUBA can be used for identifying events of relevance in recorded videos and make a time-stamp of the event, so the interesting events can be processed manually afterwards [10].

The advantage of RUBA compared to manual registration is the absence of time-consuming screening of video frames – especially in case of detection of rare events as e.g. traffic conflicts or red-light driving. As most studied cases are somehow unique, it has so far not been possible to estimate the reduction of time use for the video analyses, but it is significant.

RUBA is available for research work via contact to Aalborg University. It is the aim to share the software with partners in collaborative projects with the aim to use, test and develop the program. Furthermore, academia can access the program – but not the source code – to specific agreed projects. In order to gain further experiences with the software, The Division of Transportation Engineering at Aalborg University is also keen to carry out consultancy services of relevance to RUBA. In the long term, it is expected to make the software freely available to municipalities and consultants.

Figure K.1 shows the user interface of RUBA. RUBA allows a user to draw one or more fields on top of the video in the area or areas where analyses are requested. These fields are called detectors and can register whenever a road user passes the detector. RUBA makes these detections of road user(s) on the basis of colour changes in the videos pixels within the drawn detectors. Every time the colour changes in the detector field, it is assumed that a road user passes the detector, and RUBA makes a time-stamp of the event.

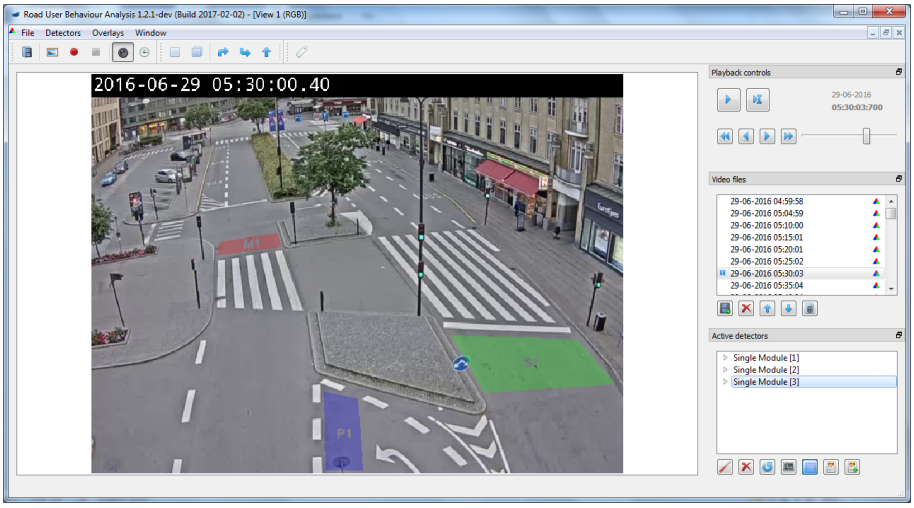


Fig. K.2: The three types of detectors in RUBA.

RUBA has three types of detectors: Presence (Blue), Movement (Red) and Stationary (Green). The Presence detector registers if a road user passes the detector area independent of the direction, the Movement detector registers if a road user passes the area in one pre-adjusted specific direction and the Stationary detector registers if a road user is standing still in the detector field. Figure K.2 shows the three types of detectors. The parameters to calibrate the detectors depend on the detector type, e.g. the parameters to calibrate a Movement detector are minimum speed, trigger threshold and movement direction, but the only parameter to calibrate a Presence or a Stationary detector is minimum occupation percentage. The minimum speed is the speed which the road user at minimum has to move to be detected, trigger threshold is the sensitivity of the detector, movement direction is the direction in which the road users should be driving to be detected and the minimum occupation percentage is the minimum coverage of the detector to activate it. The sensitivity of the detectors can be calibrated so only road users are activating the detector while movements of the camera view or branches and leaves will not affect the detection.

The detectors in RUBA can also be combined even if it is not the same type of detectors. A combination of two detectors is called a double module (One detector is called a single module). These double modules could, for example, be used to find events with same time arrival of two road users. RUBA allows the user to create more than one double module or single module in an analysis, but the amount and sizes of the detectors are crucial for how long an analysis will take. This is because it requires more pixel treatment and hence computer capacity. Figure K.3 shows an example of a double module with

Fig. K.3: Double module to detect same time arrival between a straight going bicycle and a right turning car and a straight going bicycle and a left turning car.

two Movement detectors to register cases with identical time of arrival of a straight-going bike and a right-turning car.

RUBA can be used in a lot of different analysis of traffic-related problems, and at Aalborg University so far RUBA has been used for TCS, counting traffic flows, registration of vehicle speeds and driving behaviour studies [19].

As RUBA is under development despite significant use and ongoing improvements, there are still some challenges to take into account. One is that the colour of the videos pixels in a detector field can change without a road user passes through, e.g. if the weather changes, the light in video changes or shadows from example trees or lamppost interferes. In such cases can RUBA in some cases make a time-stamp of a false-positive event regardless of the actual situation.

4 RUBA use cases

RUBA can be used for different traffic analyses, and three examples of how RUBA can be used is counting traffic flows, registration of same time arrival between road users and registration of the speed of vehicles.

4.1 Counting traffic flow – Case study in Aarhus

RUBA can count traffic flows and volumes. Traffic flows were counted in the City of Aarhus, Denmark through a zebra crossing near the central train station. The study included the cars, buses and trucks, but in this area, there

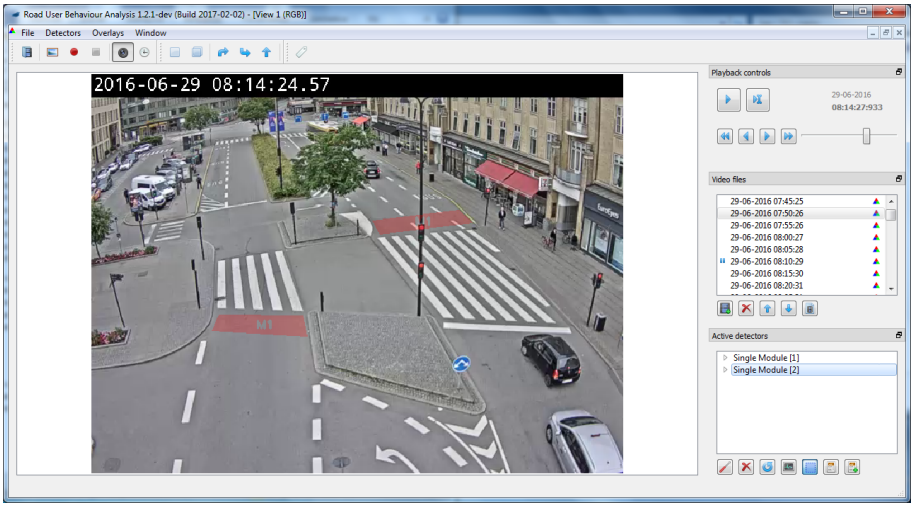


Fig. K.4: The two movement-detectors in the Aarhus project.

are many bicycles too, which means that the calibration of the detector needs to be done carefully or the detector will detect all the crossing bicycles as well.

Studies counting road users in a specific direction often use the Movement detector, because it is possible to sort out road users going in a different direction. If e.g. the straight-going cars should be counted, but the right-turning and straight-going cars share lane, the Movement detector can be used to sort out right-turning cars. To count the three types of road user, mentioned before, in this study the Movement detector is used cf. figure K.4.

The detectors are placed after the zebra crossings in the two directions so road users yielding for the pedestrians can still obtain the minimum speed to activate the detectors. In Aarhus, it was needed to be sure that the detector did not count bicycles. It was done with the parameters, minimum speed and trigger threshold, as car drivers usually drive faster and the detectors would have to be less sensitive. Figure K.5 shows an example where a car but not a bicycle is detected.

4.2 Registration of same time arrival of road users – Case study of crossroads solutions for bicycles

Another project in which RUBA was used was Road crossing for bicycles. The project focused on bicycle safety in road crossing and the effects from different kinds of bicycle lanes/paths [11, 12].

Specific modules for RUBA were developed to decrease the numbers of false-positive registrations in this study. These modules were as follows: 1: one for straight-going bicycles, 2: for right-turning cars and 3: for left-

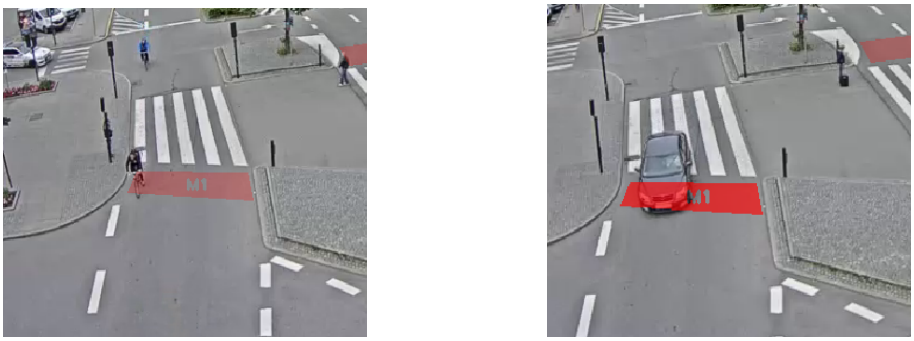


Fig. K.5: Demonstration of the ability to only detect cars, trucks and buses, and not bicycles.

turning cars. Each of the three modules contains of at least four detectors. Figure K.6 shows an example of the use the three modules. The module for straight-going cycles has four detectors: three Presence detectors (blue) and one Movement detector (red). The module for right-turning cars has five detectors: three Presence detectors (blue), two Movement detectors (red). There is one Stationary detector, which detects when a car is stationary in the detector (green). The module for left-turning cars has six detectors: two Presence detectors and four Movement detectors.

The combination of detectors used in this analysis were rather advanced and could be done more simply if more false-positives were allowed. The simpler version to detect same arrival time between a straight-going bicycle and a right-turning car would be to use a double module of two Movement detectors and a double module of a Movement detector and a Stationary detector, so it would also detect if the car yields for the bicycles. For straight-going bicycles and left-turning cars, the simpler version would be a double module of two Movement detectors. Figure 3 illustrates the principle behind the simpler version.

4.3 Registration of a vehicles speed – Case study of speed through a roundabout

A third study examined how rebuilding of rural single-lane roundabouts to road trains affects driving speed in the roundabouts [19]. The study focused on the speed of private cars and was formed as a before-after study. The rebuilding of roundabouts was made to ensure sufficient space for road trains and was mainly made from a reduction of the central island, and the reduced area was added to the driving area due to an increasing width of the lane. Hence the turning radius became bigger. A bigger turning radius could allow road users, especially private cars, to increase their speed through the roundabout. The speed through the roundabouts was estimated in RUBA

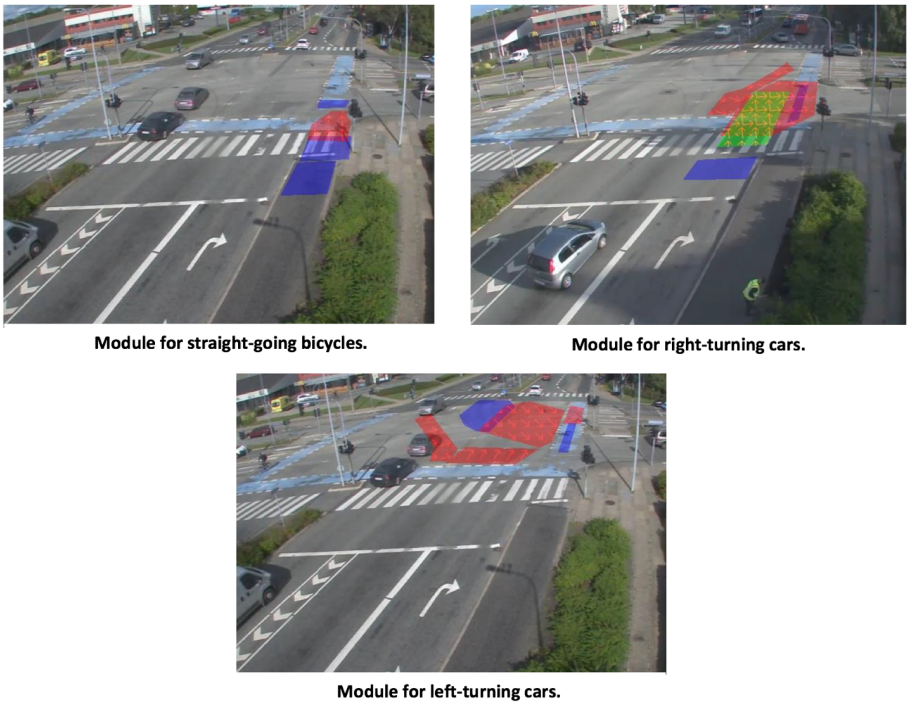


Fig. K.6: Examples of how to use the three modules.

Fig. K.7: Detectors from the vehicle speed study, before the rebuilding.

with two Single modules. The detectors registered when the car enters and leaves the detectors. With the known distance between the detectors, the speed was calculated. The overall result was that the average speed in the studied roundabout went up with 9 km/h [19]. Figure K.7 shows the detectors from one of the roundabouts. It has to be mentioned that while the actual speed registered with this method is rather uncertain, the calculated speed changes are more reliably as the same uncertainties in the before and after situation are present. The challenge regarding absolute speeds is connected with the inclined angel of recording. It is difficult to define the exact position of the detector areas and the various shapes of different cars etc. can further contribute to this uncertainty.

5 Summary and concluding remarks

Traffic safety problems are mainly concentrated where there is interaction between road users, i.e. intersections of various types. Consequently, significant parts of the focus have been on the traffic safety of these intersections. Over time it has become clear that traditional accident records are rarely comprehensive and also ethically problematic to use due to the amount of time it takes from data collection about an identified safety problems until a response. Therefore, surrogate measures are developed to react on the basis of a more comprehensive data set and in a short time. One of these measures is the Traffic Conflict Study (TCS). However, TCS requires a significant amount of video recordings to ensure coherent and reliable data samples. Analyses of this video data are very time-consuming without any software to reduce the manual work load, and it is preferable to do it more unambiguously than a human observer can do. Most available tools for these analyses are connected

to the recording camera hardware.

In order to open up for more coherent analyses and increased knowledge of the effect of various road designs on road user behaviour, a platform-independent software can contribute positively in these directions. The RUBA software offers this and can be used to analyse in TCS. Also, it has proved useful to other types of traffic behaviour studies and registrations. RUBA works as a watchdog and basically it registers when the colour pattern in a part of the video frame changes more than a defined threshold. Furthermore, it operates with three type of detectors: Presence, Movement and Stationary. The first detector records if a road user passes the detector independent of the direction, the second detector if a road user passes the area in one pre-adjusted specific direction and the third detector if a road user is standing still in the detector area.

The RUBA software has been used for various analyses of traffic behaviour: traffic counts with and without removal of different modes of transportation, traffic conflicts, traffic behaviour for specific traffic flows and modes and comparisons of speeds on rebuilt road locations. With further development of the software and the associated expected increase in server capacity and computing power, it is expected that RUBA will be an even more efficient tool than it is at present. At the time of writing, automatic detection of red-light driving is being tested. Also, at present, the first test on drone recordings has shown promising results.

There is still room for further improvement, and there is a range of challenges which should be dealt with in order to increase the benefit from RUBA or similar watchdog-based software. Some of the main challenges are the following: 1) to deal with inevitable noise from shadows, changing light conditions, unstable camera installation and other movable objects such as leaves or even birds, 2) to distinguish a detailed movement pattern, as especially for cyclists' body language and eye contact play important roles and 3) to further streamline the working procedure to reduce manual time use. These challenges could be partly met by bigger computing power. It allows, all things being equal, for the use of more detectors without using too much computer time. Also, elements of machine learning would be able to solve some of the raised issues.

Regardless of the mentioned shortcomings from such software tool, RUBA is still more unambiguous and cheaper than if a human observer is to identify various problems in the transport system. Hence, it is a highly relevant tool to contribute to applied field research and, as such, to a cleaner, safer and more efficient road transport system in the future.

Acknowledgement

The authors would like to thank the InDeV project for the overall contribution to the software development. Also, special thanks to the Municipality of Aarhus, Denmark for their interest in this upcoming technology and their interest in measuring the effects of new road and traffic signal designs.

References

[1] "Indev - in-depth understanding of accident causation for vulnerable road users."

[2] N. H. T. S. Administration *et al.*, "Early estimate of motor vehicle traffic fatalities for the first 9 months of 2016," *DOT HS*, vol. 812, p. 358, 2017.

[3] N. Agerholm and C. S. Andersen, "Accident risk and factors regarding non-motorised road users-a central road safety challenge with deficient data," *Latin American Journal of Management for Sustainable Development*, vol. 2, no. 2, pp. 102–111, 2015.

[4] COWI, "Trafikregistrering ved brug af droner (traffic registration from drones)," http://www.cowi.dk/menu/service/oekonomimanagementogplanlaegning/trafikplanlaegningogmodellering/trafikplaner/trafikregistrering-ved-brug-af-droner/, 2015.

[5] DataFromSky, "Video analysis," http://datafromsky.com/services/video-analysis/.

[6] C. Hydén, "The development of a method for traffic safety evaluation: The swedish traffic conflicts technique," *Bulletin Lund Institute of Technology, Department*, no. 70, 1987.

[7] Iteris, "Pedtrax," http://www.iteris.com/products/pedestrian-and-cyclist/pedtrax, 2017.

[8] S. Jackson, L. F. Miranda-Moreno, P. St-Aubin, and N. Saunier, "Flexible, mobile video camera system and open source video analysis software for road safety and behavioral analysis," *Transportation research record*, vol. 2365, no. 1, pp. 90–98, 2013.

[9] A. Laureshyn, "Application of automated video analysis to road user behaviour," 2010.

[10] T. K. O. Madsen, P. M. Christensen, C. Bahnsen, M. B. Jensen, T. B. Moeslund, and H. S. Lahrmann, "Ruba-videoanalyseprogram til trafikanalyser," *Trafik and Veje*, no. 3, pp. 14–17, 2016.

[11] T. K. O. Madsen and H. Lahrmann, "Krydsløsninger for cyklister: Anvendelse af konfliktteknik til vurdering af forskellige løsningers sikkerhed," 2014.

[12] ——, "Comparison of five bicycle facility designs in signalized intersections using traffic conflict studies," *Transportation research part F: traffic psychology and behaviour*, vol. 46, pp. 438–450, 2017.

[13] E. R. S. Observatory, "Traffic safety basic facts 2015 - main figures," *Brussels: European Commission, Directorate General for Transport*, 2016.

[14] S. M. of Enterprise and Innovation, "Renewed commitment to vision zero is now being launched – for improved transport safety," 2016, stockholm.

[15] W. H. Organization, *Global status report on road safety 2015*. World Health Organization, 2015.

[16] P. St-Aubin, "Driver behaviour and road safety analysis using computer vision and applications in roundabout safety," Ph.D. dissertation, Ecole Polytechnique, Montreal (Canada), 2016.

[17] Å. Svensson and C. Hydén, "Estimating the severity of safety related behaviour," *Accident Analysis & Prevention*, vol. 38, no. 2, pp. 379–385, 2006.

[18] V. T. Systems, "Traffic flow," http://www.viscando.com/traffic-flow, 2015.

[19] C. Tønning and N. Agerholm, "Speed changes in rural single-lane roundabouts converted for road trains," 2018, forthcoming.

Paper L

Collecting Traffic Video Data using Portable Poles: Survey, Proposal, and Analysis

Morten B. Jensen, Chris H. Bahnsen, Harry S. Lahrmann, Tanja K. O. Madsen, and Thomas B. Moeslund

The paper has been published in the
Journal of Transportation Technologies Vol. 8 No. 4, pp. 376–400, 2018.

The layout has been revised.

Abstract

Several initiatives have been launched to help prevention of traffic accidents and near-accidents across the European Union. To aid the overall goal of reducing deaths and injuries related to traffic, one must understand the causation of the traffic accidents in order to prevent them. Rather than deploying a person to physically monitor a location, the task is eased by camera equipment installed in existing infrastructure, e.g. poles, and buildings, etc. In rural areas there is however a very limited infrastructure available which complicates the data acquisition. But even if there is infrastructure available in either the rural area or the urban area, this might not serve as an ideal position to capture video data from. In this work, we survey and provide an overview of available and relevant portable poles setups with respect to capturing data in both urban areas and rural areas. The conclusion of the survey shows a lack of a mobile, lightweight, compact, and easy deployable portable pole. We therefore design and develop a new portable pole meeting these requirements. The new proposed portable pole can be deployed by 2 persons in 2 hours in both rural areas as well as urban areas due to its compactness. The deployment and usage of the new portable pole is a complimentary tool, which may improve the camera capturing angle in case existing infrastructure is insufficient. This ultimately improves the traffic monitoring opportunities. Further, the survey of selected portable poles provides an excellent overview and can aid multiple applications within road traffic.

1 Introduction

Preventing traffic accidents and near-accidents remains a major and interesting challenge to address for academic partners as well as public organizations. In 2017 alone, the European Union (EU) reported that 25,000 people lost their lives and 135,000 people were injured on the roads across the EU [5]. In 2009 the EU estimated that the deaths and injuries across Europe costed the society approximately 130 billion Euro [4]. As a result, the EU set out a 2010-2020 goal with an overall objective of halving road deaths across Europe. To achieve this, several initiatives have been started covering increased enforcement of road rules, improved education and training of road users, safer road infrastructure, promote the use of modern technology to increase road safety (ITS), and protection of vulnerable road users (VRU). All of which are important to analysis and address to meet the overall objective in 2020.

Understanding accidents causes in the traffic requires a lot of data, which can be collected with different purposes. Naturalistic Driving Study (NDS) such as the "100-Car Naturalistic Driving Study" [16] and the "SHRP2 Naturalistic Driving Study" [3], collects all sorts of data from within the participating vehicles such as GPS, accelerometer and similar vehicle network data, but the

vehicles are also equipped with multiple different sensors, e.g. RGB cameras, thermal cameras, stereo cameras [18] or radars. Though these studies generate a lot of interesting data, a major drawback of this approach is the large investments needed to reach a large participant pool and then afterwards installing expensive equipment inside the car whilst keeping the car naturalistic.

A less expensive approach of capturing data that helps understanding accidents causes is simply to monitor and observe a critical location, e.g. traffic intersection. This manually task is however quite error-prone as the assigned person must be aware of everything happening in area of interest whilst continuously documenting the observations over a longer period of time. So rather than deploying a person to physically monitor a point of interest, the task is eased by mounting a camera-based system in existing infrastructure, e.g. poles, and buildings, etc. The captured video data can then be post-processed and analyzed with the purpose of understanding the scene and ultimately making adjustments that ideally prevents accidents and near-accidents. The main challenge of the camera-based system is that often there is no or very limited existing infrastructure available at the scene, thus directly impacting the quality of the analysis. This has spawned the use and interest in portable setups that can be moved around, which allows for a more optimal data collection in both urban areas but in particularly also in rural areas where there is often no proper infrastructure to mount cameras in.

In this paper, we make an analysis of relevant portable setups, where we discuss the pros and cons of different portable types and solutions, thorough overview of available setups. The result of the overview shows a lack of a mobile, lightweight, and easy deployable portable pole, thus we design and develop a new portable pole meeting these requirements.

The contributions of this paper are thus twofold:

1. Providing a thorough analysis and overview of available portable camera-based capturing setups.
2. Design and development of a new mobile, lightweight, and easy deployable portable pole to ease camera-based data collection.

The paper is organized as follows: Section 2 describes the minimum requirements for the portable pole as well as the general definitions used. All of the requirements and definitions are then used examining various solutions ultimately providing an overview of available portable pole solutions in Section 3. In Section 4, the design and development of the new portable pole is presented. Usage and applications of new portable pole is presented in Section 5. In Section 6 we perform a discussion of our work. Finally, we present our conclusions in Section 7.

2 Portable Pole Analysis

Portable poles can serve multiple purposes and can be used for various applications. As briefly mentioned and introduced in Section 1, this survey will only consider portable pole solutions that could be relevant as a camera-based recording platform in the field of traffic surveillance and monitoring.

2.1 Minimum Setup Requirements

The relevant portable pole solutions are derived based on 4 minimum requirements that are considered essential for a portable pole to function as a proper camera-based recording platform, which can be utilized in both urban and rural traffic environments.

Recording Time

The video recordings are the basis for the entire analysis, so besides having a great view-angle provided by either the infrastructure or a portable pole, the video recordings must contain a sufficient amount of accidents or near-accidents in order to make some concluding remarks of a given location. In [7], the frequency of traffic accidents is described as a pyramid, where the pyramid base contains normal traffic encounters that are non-critical and rather safe, but very frequent. The pyramid apex contains the fatal and very severe events, e.g. fatal injuries, these are however occurring more infrequent compared to accidents in the lower part of the pyramid. Previous studies from Scandinavia show that at a particular site, the number of near-accidents tends to be as low as 1-2 per day [6] [11] [20]. So in order to get video recordings containing some infrequent events, the portable pole and camera-based setup must robust and stable enough to record continuously throughout a longer period of time. In this analysis we consider a period of 3 weeks to be the minimum requirement.

Capturing Height

A major issue to take into account when installing camera equipment at a point of interest is occlusion. Occlusion is in this case defined as when two objects are overlapping each other from the view-angle of the camera equipment, which makes the objects completely or partly occluded. In Figure L.1 an example of this is shown, where the red car is clearly not visible from the specific camera-view mounted in existing infrastructure.

To reach the most accurate conclusion in a traffic analysis, the data needs to be as accurate as possible, thus we want as little occlusion as possible in the data collection. There are multiple ways of reducing occlusion, e.g. having multiple cameras from different view-angles or simply just by increasing the capturing height similar to the Figure L.1b. In this analysis, we define

(a) (b)

Fig. L.1: Objects can overlap each other in the camera-view as seen in (a) where the the large cement truck clearly occludes the lane behind it. (b) clearly shows that a red car is in fact driving side-by-side of the cement truck.

a minimum capturing height of 7 meters for the portable pole, which is 3 meters higher than the maximum height limit for vehicles in most countries in Europe [21] [8].

Ground Area Occupation

To make sure, that the data collection is done in an as naturalistic and unobtrusive environment as possible, we need to make sure that the base does not cause any major impact on the behavior of the drivers on the road or the pedestrians on the sidewalk. Naturally, placing a new "intruder" in an existing environment may attract some attention and thus result in changed driver behavior, but the point of this demand is to keep it at a minimum by defining the maximum ground area occupation of the portable base to be 1.5 meters in the width. This should enable deployment of the portable pole in rural areas and in most urban environments as it can be deployed on the sidewalk whilst pedestrian should be able to easily walk around it. The maximum ground area occupation is only defined for the width, as this is the strictest one in terms of occupying the sidewalk. The length is less critical as people are still able to use the sidewalk, however it should preferably be under 2.5 meters.

Payload Weight

The portable pole setup must be able to handle the payload weight from the capturing devices mounted in the top. In this analysis, we suggest using both a RGB camera and thermal camera as capturing devices. Using multi-modal visual cues provides a solid data foundation for a later accident causation analysis as accidents and near-accidents do not solely happen in daylight [19]. Doing periods with a limited amount of light and challenging weather conditions, e.g. night, winter, rain. Thermal cameras are quite useful as

illustrated in Figure L.2, where both modalities are seen showing the same scene.

(a) (b)

Fig. L.2: Data collection at 02:00 in the night using two modalities: (a) RGB camera (b) Thermal camera.

The RGB camera is having a hard time coping with the headlights from the car and the low-light in the reminder of the scene. Furthermore, the RGB camera seen in Figure L.2a is challenged by the weather conditions, i.e. rain. The thermal camera on the other hand do not rely on light to produce its output but infrared radiation, which clearly produce a more accessible output as seen in Figure L.2b, where the car is clearly visible. The pole must therefore be able to handle a setup with two capturing devices. The capturing devices in this analysis are seen in Table L.1, which defines a minimum payload weight requirement of 5.7 kg.

Table L.1: Derivation of the minimum payload weight requirement using AXIS RGB camera and thermal camera.

Type	Manufacturer	Model	Weight [kg]
RGB	Axis	Q1615-E	3.5
Thermal	Axis	Q1932-E	2.2

Below are the requirements for a portable pole listed, if nothing else is stated, these are minimum requirements.

1. Solution must be able to record continuously in 3 weeks.
2. Capturing height: 7 m.
3. Maximum ground area occupation(Width): 1.5 meters.
4. Payload weight: 5.7 kg.

2.2 Portable Pole Types

In this analysis, we have divided portable poles into 4 different types, which will also form the structure for the reminder of the portable pole analysis and overview, namely: 1) Lightweight and compact portable pole with low payload; 2) compact portable pole with high payload; 3) trailer portable pole with high payload; and 4) heavyweight portable pole with high payload.

The payload is the capacity which the portable pole is able to lift in the top during operation. The stability in the top of the pole, hence the recording usage quality, is dependent on the payload. Common for all of the portable pole types are that they all must comply with the minimum requirements defined in Section 2.1.

Type-1 ***Lightweight and compact portable pole with low payload:*** The main goal of this type is that they are very easily moved and transported between locations. The efforts needed for setting up this type of portable pole is very low. The setup and transportation of this type of portable pole is a one-person job, requiring it to be lightweight and compact. The stability and payload scales accordingly, resulting in a low payload to keep the pole stable in the top.

Type-2 ***Compact portable pole with high payload:*** Rather than being able to transport the portable pole by yourselves, this type consider more heavy-weight equipped that can be assembled on-location by one or two persons. The equipment will remain compact while dissembled such it can be easily transported from location to location by use of a van or pick-up truck. When assembled the equip-ment is more robust compared to type-1, but at the cost of easy mobility.

Type-3 ***Trailer portable pole with high payload:*** This type utilizes a trailer or small wagon which can be attached to a vehicle's hitch ball. All the equipment is installed upon this trailer, such that one or two persons can drive to a location and set up the portable pole without too much assembling and more lenient requirements for the level of the ground base. This provides a rather stable portable pole with some degree of mobility.

Type-4 ***Heavyweight portable pole with high payload:*** By using a large platform of e.g. concrete, all the equipment can be installed on this providing a robust platform for the portable pole. However, this require a large truck with a crane for transportation, but provides a good pre-assembled portable pole.

This division will form the structure for the portable pole overview section when surveying the corresponding available portable poles.

3 Overview of Relevant Portable Poles

The overview is divided into 6 parts. The first part introduces a general base framework that complies with the battery and storage requirements and is applicable for most of the portable poles presented. This is followed by 4 parts, one for each of the 4 portable pole types presented in Section 2.2. The final part presents an overview that summarizes all of the presented portable poles.

3.1 Base framework

Regardless of the portable pole choice, the data recording capacity, the power supply and underlying video acquisition framework must fulfill the minimum requirements. Using aforementioned minimum requirements, we will in this subsection define a common framework that can be used together with the portable poles.

Video acquisition

The Axis cameras defined in Table L.1 are capable of operating by Power over Ethernet (PoE) which means it is only necessary to supply one cable per camera in the mast. The cameras are by the use of a network switch connected to a Synology DS215j Network Allocated Storage (NAS) server, where the acquired video data must be properly stored. The storage capacity required is heuristically derived to be no less than 6 TB in order to keep 3-weeks of data using H.264 compression.

Power supply and enclosure

The video acquisition hardware presented above must be powered throughout the 3-week acquisition period. The power supply and some of the video acquisition hardware must also be placed in an enclosure which is resistant to tampering.

The video acquisition hardware consumes approximately 30 watts in operation, which make a self-contained setup unfeasible due to 3-weeks video acquisition requirement. Instead we use 3 heavy-duty 12 volt 180 Ah batteries, which provides the setup with an approximately replacement cycle of 4-6 days depending on the overhead and wear out of the batteries. The entire system is finally installed in an IP65-certified Eurobox 40705, which can be seen in Figure L.3.

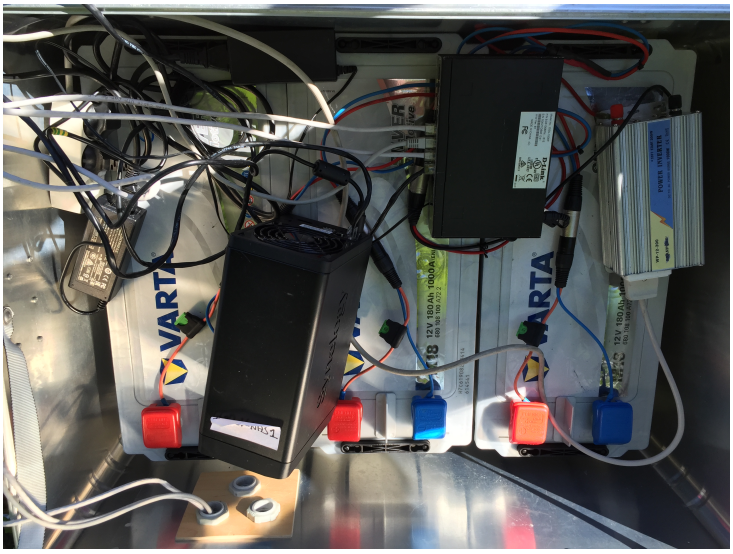


Fig. L.3: The Eurobox 40705 containing 3 batteries, a 230V power inverter, a PoE switch, and a NAS server.

3.2 Type-1: Lightweight and compact portable pole with low payload

The first type of poles, is as introduced in Section 2.2, the most compact and lightweight ones, and should ideally be deployable for a single person.

Miovision Scout

Scout is a portable and expanded pole developed by Miovision, and is, according to their own documentation, "designed specifically with the users in mind" [15]. This has resulted in a portable pole with a weight of only 19.1 kg and a set up time of 10 minutes. The Miovision Scout do not meet the requirements for this analysis, defined in Section 2.1, as it is not configurable for the two cameras defined in Table L.1. It is however still included as it is a very popular solution for traffic monitoring, and might be usable in pilot tests or as a second view-angle.

The Miovision Scout has a battery life of 7 days when buying the additional power pack and can be set up on existing infrastructure using an included pole mount. The simplicity of the product can easily be deducted by examining Figure L.4. In case deployment is needed in places without street poles, a separately sold Scout Tripod can be used. The Scout Tripod weights 14 kg, but can reach 68 kg with additional security weights. The Miovision Scout is equipped with a wide lux camera with 120°horizontal view capturing with a

Fig. L.4: The lightweight and compact, but non-configurable, portable pole from Miovision. (a) Miovision Scout with the extendable pole [15] (b) Miovision Scout Video Collection Unit. [15]

resolution of 720x480 pixels @ 30 FPS. As mentioned in the introduction, this camera setup is not configurable. The operational height can be adjusted to be between 1.32-6.4 meters, which do meet the requirements either. In Table L.2 an overview of the required equipment is seen.

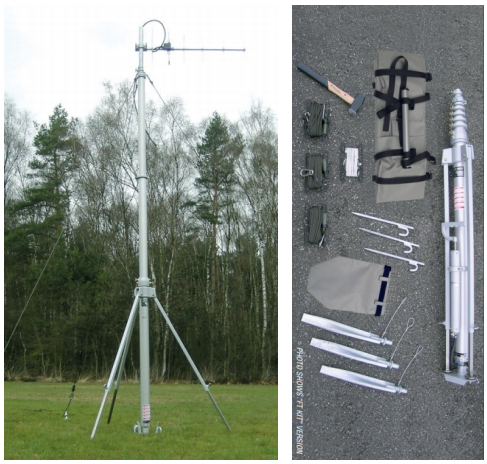
Table L.2: Required equipment for the Miovision Scout.

Product	Weight
Scout Video Collection Unit	10.89 kg
Scout Pole Mount	8.16 kg
Scout Power pack	14.0 kg
Scout Tripod	14.0 kg

The Miovision Scout can be mounted to existing infrastructure, such as a pole, defining some requirements to how poles or similar objects are located at an intersection. Otherwise the Scout Tripod can be used to deploy the Scout. For both of the solutions no major equipment is needed, and one person should be able to set this up in an hour.

Custom lightweight portable pole

This portable pole is a proposal on how a lightweight portable pole could be manufactured. The portable pole must meet the requirements defined in Section 2.1, while being a lightweight solution easily transported around.



(a) (b)

Fig. L.5: Parts for a custom lightweight portable pole (a) Clark Mast FT mast. (b) Clark Mast FT carrying bag. [12]

The portable pole utilizes the Clark Masts SFT9-6 mast, which can be extended to 8.8 meters using a hand pump, and remains at 2.05 meters in retracted mode. An image of a FT series mast from Clark masts is seen in Figure L.5a. On top of this there must be created a rig which cameras can be mounted in. The mast comes with a carrying bag, seen in Figure L.5b for easier transportation. In addition to the bag, equipment such as spikes and radius lines are also included. This solution must also utilize the base framework presented in Section 3.1. In addition to the base framework, Table L.3 summarizes the additional required equipment to manufacture this type of pole.

Table L.3: Equipment needed for custom lightweight portable pole. An unknown weight is marked with a "-".

Type	Model	Weight
Telescopic mast	Clark Masts FT series, SFT9-6/HP 10 kg headload, 8.80 m extended height, 2.05 m retracted height, w. tripod	-
Carrying bag	Clark carrying bag, SFT9-6/HP Bag	-

A van must be used to transport the equipment from location to location as the mast is 2.05 meters long, but setting up the equipment should be doable for one person. Using the radius line to make a guying system is however not really feasible in urban places, requiring the wind speed to be low for the

setup to remain usable.

Discussion

The Miovision Scout do not meet the configurable requirements defined in Section 2.1 and can therefore not be used in the final setup. It might, however, be a useful solution for some minor pilots tests or be used a second view-angle at a complex environment. The custom made portable pole is not as lightweight as the Miovision Scout as one needs to bring more equipment to meet the requirements of capturing data continuously in 3 weeks. The custom made portable pole can be configured to have 2 cameras installed, but it is however considered necessary to utilize a guying system in order to stabilize the portable pole sufficiently, even in low wind conditions, such the video recordings are stable and usable for a traffic analysis.

3.3 Type-2: Compact portable pole with high payload

We divide possible solutions for systems using a compact portable pole with high payload into three proposals based on the estimated total weight of the system: lightweight, middleweight, and heavyweight. All of them utilize the base framework presented in Section 3.1.

Lightweight: Mast with tripod

The lightweight portable solution consists of a telescopic, 5-section mast with a corresponding tripod. The extended mast is usually secured by a guying system to assure stability under heavy payload and wind speeds. However, as guying is not applicable in urban areas, we include a tripod to ensure stability. The tripod furthermore ensures independence of existing infrastructure and comes in a variety of sizes for different mast heights. An image of such a pole is seen in Figure L.6.

We choose the largest mobile tripod available to provide stability and accommodate the requirements even under moderate wind speeds and payloads. The base diameter of the tripod is 2 m, which have a recommend maximum mast height of 10m. When the mast is not guyed, the maximum wind speed is 13.8 m/s for stable operation. A wind speed of 13.8 m/s translates to 'Strong Breeze' on the Beaufort scale.

The necessary equipments for the lightweight mast with tripod are listed in Table L.4.

The transportation of the equipment requires a medium-to-large sized car or van to accommodate the length of the retracted mast and the total weight of the equipment. The telescopic mast is extended by an integrated hand pump, and the extended section is subsequently locked manually by using the provided screws. The ground area required for the base is 0.5 m larger than

Fig. L.6: Clark QT Mast on tripod [14].

Table L.4: Required equipment for compact, lightweight portable pole with high payload. Maximum wind speed 13.8 m/s. An unknown weight is marked with a "-".

Product	Model	Weight
Telescopic mast	Clark QT Series, SQT9-5, 5 section mast 18 kg payload, 9.00 m extended height, 2.25 m retracted height	-
Tripod	Clark MK VI, MK6 2000MM	18.0 kg
Tripod adapter	Clark	-

specified in the setup requirements. The extra space is however necessary for the stability of the portable pole.

Middleweight: Mast with tripod

The lightweight setup, described in Section 3.3, is used as a point of departure for the middleweight portable setup where the telescopic mast and tripod remain key components. The Clark QT mast from the lightweight setup is replaced by the heavier and sturdier NT series and features only 4 sections compared to the 5 section QT mast. The heavier mast calls for a heavier and larger tripod which is found in the Clark MK IV Tripod. The tripod weighs 27 kg and features a base diameter of 2.6 m. As with the light-weight mast with tripod, the un-guyed mast is stable up to wind speeds of 13.8 m/s. The equipment of the middleweight mast with tripod is listed in Table L.5.

Due to the larger retracted height of the telescopic mast (2.82 m) it might be impossible to fit inside an ordinary car, and thus a larger van is recommended.

Table L.5: Required equipment for compact, middleweight portable pole with high payload. Maximum wind speed 13.8 m/s. An unknown weight is marked with a "-".

Product	Model	Weight
Telescopic mast	Clark NT Series, NT 90-4, 4-section mast, 15 kg payload, 9.00 m extended height, 2.82 m retracted height	41.0 kg
Tripod	Clark MK IV	27.0 kg
Tripod Adapter	Clark	-

The telescopic mast is extended by the use of a hand pump and the sections are secured by screws similarly to the lightweight setup. The ground area required is even larger than for the lightweight scenario; however, this is needed in order to provide stability for the heavier mast.

Heavyweight: Flyintower

The heavyweight compact portable pole solution uses a Flyintower, or sound tower, as the camera mast. The Flyintower is a well-known object at large concerts or festivals where it is used for the lifting of loudspeakers as depicted in Figure L.7. The V-shaped basement, the metal grid, and the heavy weight of the construction improve the sturdiness and stability of the setup.

Fig. L.7: Litec 7.5-500 Flyintower [10].

We choose the smallest possible Flyintower from Litec to minimize the ground occupation area required for the basement of the tower. The extended height of the tower is 7.75 m and due to the V-shaped basement, the footprint

is 4.1 × 3.6 m. The maximum lifting load capacity of the tower is 500 kg which requires additional ballast at the base for stability. For the much lighter loads required in this setup, the required ballast weight is reduced. Due to the studier nature of the setup, the maximum wind speed is increased compared to the lightweight and middleweight setups. A list of the equipment is found in Table L.6.

Table L.6: Required equipment for compact, heavyweight portable pole with high payload. Maximum wind speed 70 km/h. An unknown weight is marked with a "-".

Product	Model	Weight
Flyintower	Litec 7.5-500, 500 kg max load capacity, 7.75 m extended height	160 kg
Ballast	Required ballast for Flyintower	-

The Flyintower is considerably heavier than the mast-based solutions listed above. However, the tower might be taken apart and assembled on-site which greatly reduces the space needed for storage and transportation. We therefore estimate that a larger van is needed for the transportation, just as in the middleweight scenario. Compared to the lightweight and middleweight scenarios, the Flyintower requires a larger, planar surface for the base to stand. This might exclude the deployment in tight urban spaces where such space is not available.

Discussion

For both the light portable poles and middleweight portable poles, issues arise when dealing with higher wind speeds as the equipment is mounted in the top making the setup unstable. To cope with this, a guying system can be installed to stabilize the mast, this is however not feasible in urban places. For most scenarios in urban environments, both portable pole setups are considered usable in terms of wind speeds. The heavyweight solution is therefore a better overall option due to increased stability, but significantly comprising the compactness and weight compared to the lightweight and middleweight solutions. Generally, all of the solutions can possibly be disassembled and be somehow compact and then be used in rural areas where there are more open space, it is however not ideal that none of the proposals meet the maximum ground occupation area requirement. Deploying any of the introduced solutions in this section in an urban environment will most likely be considered unnaturalistic and obtrusive.

3.4 Type-3: Trailer portable pole with high payload

The third type of portable poles differs from the both type-1 and type-2 in the sense that the equipment used comes in a more wrapped up and easy-deployable way. As mentioned in Section 2.2, type-3 relies on equipment installed either in a trailer or in a small wagon resulting in less assembling on-site.

UTRaCar

The Urban Traffic Research CAR is developed for the national aeronautics and space research center of the Federal Republic of Germany (DLR) [1] and is equipped with a large set of sensors and systems to be used for traffic surveillance and data acquisition in the field. The car is seen in Figure L.8a in transportation mode and in Figure L.8b where the left image show an image of the car in operation [9]. The UTRaCar does not meet the requirement of the maximum ground occupation area, but is included as it provides some interesting solution ideas.

(a) **(b)**

Fig. L.8: (a) The DLR UTRaCar with retracted telescopic mast. (b) The DLR UTRaCar with extracted telescopic mast. [9]

The car is equipped with multiple sensors as seen from the images in Figure L.8b. For this analysis, the telescopic mast seen in the left image is the most interesting one. A telescopic mast is mounted in the back of the car, and can extend to 13 meters. In the top of the telescopic mast various sensors can be installed, as seen in the upper right image in Figure L.8b. According to [14], the power supply unit in the car is self-sufficient. It is unclear what this covers, but from the lower image in Figure L.8b, it is clear that a lot of

equipment can be installed in the of the car. In Table L.7 an estimate of the equipment needed for a minimum requirement solution are seen.

Table L.7: Estimated equipment needed for a minimum requirement version of the UTRaCar. An unknown weight is marked with a "-".

Type	Model	Weight
Van	VW Crafter 35 with medium wheelbase and high roof	-
Telescopic mast	Clark WT Series, WT100-4, 4 Section mast, 140 kg headload, 10.0 m extended height, 3.32 m retracted	-

The size of the car can be a challenge at a lot of intersections, so there must be some open areas around the intersection for deploying this system. But if the area suffices, a solution like this allows a rather fast deployment without any external actors.

Trivector Mobile Mast

The Swedish based company Trivector has developed the TMV1, which is a mobile mast installed in a trailer with the scope of capturing traffic situations. When extracted the height can reach up to 15 meters. In Figure L.9a an image of the setup is shown, and in Figure L.9b it is visible that the setup utilizes two cameras in operation meeting the requirements for this analysis.

The setup consists of a trailer equipped with a custom made telescopic mast. Inside the trailer all the equipment can be stored, and given from the image seen in Figure L.9a, it is clear that box is rather large, providing good possibilities to put all equipment inside. There exists no technical data sheet available to the public, hence it is hard to estimate the equipment used to create the Trivector mobile mast. From examining the figures the minimum requirements are a cargo trailer and a telescopic mast. As for the UTRaCar, the setup occupies a rather large area on the ground, making it difficult to place in some urban areas. The installation complexity is low as it all equipment are inside the trailer, so the deployment is straightforward with a minimum of external actors. Finally, a car is needed to tow this setup from point A to point B.

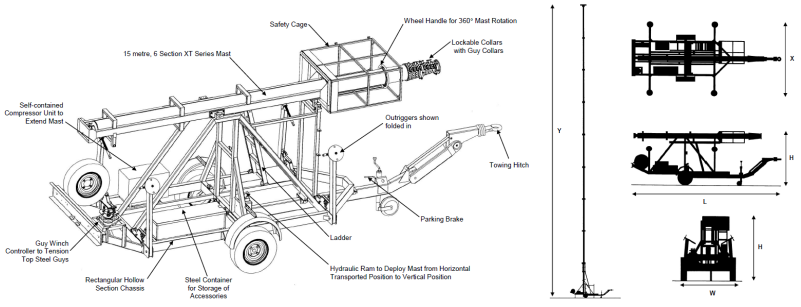
Custom made trailer

With inspiration of the previously solutions in type-3, we look into to assembling a trailer portable pole. The main idea is to utilize a trailer solution with a pole mounted on it. In Figure L.10a and Figure L.10b the main component in the setup is seen. It consists of an already existing product which needs to

(a) (b)

Fig. L.9: (a) The Trivector Mobile mast setup in operation. (b) The Trivector Mobile mast with two installed cameras. *Images provided by Aliaksei Laureshyn, Lund University.*

be customized to accommodate the minimum requirements. Though there are some boxes and containers mounted in the original Clark Mast 804-15-6, additional room is considered necessitated to meet the capacity requirements. The 6 section XT Series mast mounted on the trailer has an extracted height of 15 meters. [13]



(a) (b)

Fig. L.10: Specifications and overview of Clark Mast 804-15-6. [13]

A vehicle is needed to tow the trailer from location to location. A regular

van is considered to be sufficient to tow the trailer and the remaining equipment The length of the trailer is 6.3 meters, making it quite large and difficult to deploy in tight urban spaces.

Discussion

For all of the trailer solutions the main advantages are the easy and rather fast deployment as a very limited amount of external actors are needed. Other advantages of the type-3 solutions are a "all-in-one" solution in the sense that room for batteries, HDD and other equipment is included in the setup. The disadvantages are that they can rather fast become quite expensive, and they do not scale very well in the sense of transportation from point A to point B might require multiple cars or a large truck. The weight and size of the solutions also occupies a large ground area which challenges one of the requirements defined for this analysis. Furthermore, a regular driver's license might not be sufficient for all of the solutions. The UTRaCar solution is considered to become quite expensive to build, so the best option in type-3 is to use a solution similar to the Trivector mobile mast or the custom made trailer, even though it is also expected to become expensive.

3.5 Type-4: Heavyweight portable pole with high payload

The last type of portable poles takes it starting point in a large platform of e.g. concrete, where all equipment can be installed upon. This complicates the transportation phase but should have advantages in operation compared to the previous types.

DLR Platform

The National Aeronautics and Space Research Centre of the Federal Republic of Germany (DLR) have used a portable platform for data capturing. The development of the technical aspects of the portable platform was carried out by Jenoptik. One of the usages of it has been to monitor railroads crossings as seen in Figure L.11a. The camera equipment used in the DLR portable pole setup consist of 4 cameras, 2 IR-flashes, 2 radars, and an aluminium frame, totaling a payload weight of 25.4 kg. In operation mode, the camera fixed to an operational height of approximately 4-5 meters. [17] As seen from Figure L.11 it is clear that the entire portable pole consists of a cabinet and a port that is split into two pieces, and is mounted onto a large concrete block. In operation mode, the pole is angled in vertical position, opposite to the horizontal transportation angle seen in Figure L.11b. The equipment needed for creating a portable pole for meeting the minimum requirements are seen in Table L.8. The mast could however be changed to a telescopic mast.

(a) (b)

Fig. L.11: (a) DLR Setup in operation mode. (b) DLR Setup in transportation mode. [17]

Table L.8: Required equipment for heavyweight portable pole with high payload mounted on a concrete block. An unknown weight or model is marked with a "-".

Type	Model	Weight
Custom mast	Mast divided into two parts: Transportation and operation mode	-
Cement block	-	-
Vandalism-proofed cabinet	-	-

According to an interview with Kay Gimm and Sascha Knake-Langhorst from DLR, it can require up to a whole day to setup and calibrate the sensors for the specific application. As the current system requires power supply access from the current infrastructure. Due to that concrete block, the setup is quite heavy requiring a truck and crane to move it around.

Discussion

Only one solution is presented for the type-4, which is the DLR setup. This setup provides a good and solid platform for data capturing. Installed on the concrete block is all the equipment needed making it a "all-in-one" solution. However, the setup is heavy meaning a truck and crane is needed for transportation and deployment.

3.6 Overview

Creating a setup that is lightweight, robust, and as mobile as possible is a hard problem to satisfy. It might become easier to record traffic data at certain

intersections if using a small and lightweight setup. One can, however, not be certain of the quality of the recordings as lightweight usually correlates with instability during varying weather conditions; especially when considering that the setup has a relatively heavy camera rig mounted in the top. All of the surveyed options are seen in Table L.9.

The main parameter to satisfy is considered to be the recording quality, as the quality of the data is essential for performing a good traffic analysis. Taking this into account, the proposed solutions from both type-1 and type-2 are not good options as they require guying systems in order to reach stability for prolonged periods of time. Guying systems are not ideal in urban environments, and the lightweight and compact pole solutions examined in this analysis does therefore not pose an ideal fit for the requirements.

For both type-3 and type-4, the solutions presented will provide some more stable recording platforms however they are considered quite expensive to produce, and does therefore not scale very well. Furthermore, the solutions of type-3 are in most cases wider than the specified maximum of 1.5 meters, hampering the deployment on the sidewalk without interrupting the pedestrians. The type-3 solutions are, however, more mobile compared to the type-4 solution, but in both cases a regular driver's license might not be sufficient. Additionally, the trailer option does not scale well as multiple trailers requires multiple towing vehicles.

This leads to the conclusion that for capturing the most stable and useful data, the setup must comprise the lightweight and easy mobility requirements. For type-1 and type-2 solutions to work, various guying system must be installed on existing infrastructure to fixate the pole. If one involves the existing infrastructure, a better result would be reached if the capturing rig is mounted on the infrastructure rather than using a light-weight or compact portable pole with guying installation. The type-4 solution from DLR requires both a truck and a crane to deploy, which satisfies most of the requirements for this analysis, but remains, however, the less mobile solution in this analysis.

4 Design & Development of TRG-Pole

In this section, we will present a pole which is hybrid between a type-2 and type-4 portable pole solution designed specifically to contain the same advantages as the DLR solution while being mobile.

4.1 The designed pole

We present a portable pole design that accommodates the overall portable pole goal while being in operation mode. It is, however, desirable to keep the weight down during transportation. To reach this, we propose creating the

Table L.9: Overview of the analyzed portable poles. The poles are summarized and can easily be compared on the 7 different parameters.

Type	Type Name	Operational height [m]	Payload [kg]	Operational base dimensions [L x W x H [m]]	Transport dimensions [L x W x H [m]]	Weight [kg]	Configurable	Deployment equipment
T-1	Miovision Scout [15]	1.3 - 6.4	-	1.5 x 1.5 x 1.24	-	48	No	Car
	Custom lightweight portable pole	8.8	10	-	2.05 x 0.25 x 0.25	-	Yes	Van
T-2	Lightyweight: Mast with tripod	9.00	18	2.0 x 2.0 x 9.0	2.25 x 0.4 x 0.4	-	Yes	Car
	Middleweight: Mast with tripod	9.00	15	2.6 x 2.6 x 9.0	2.8 x 0.4 x 0.4	68	Yes	Car
	Heavyweight: Flyintower	7.75	500	4.1 x 3.6 x 7.75	-	>160	Yes	Van
T-3	UTRaCar [1]	13	-	5.9 x 2.4 x 2.4	5.9 x 2.4 x 2.4	>2800	Yes	-
	Trivector Mobile mast	15	-	-	-	-	Yes	Car
	Custom made trailer	15	140	6.3 x 1.95 x 15	6.3 x 1.95 x 2.2	-	Yes	Car
T-4	DLR Platform	4-5	25.4	-	-	-	Yes	Truck, crane

pole as a hybrid between a type-2 and type-4, meaning that the pole is compact and has a reduced weight during transportation, but which in operation mode remains robust and stable. One of the main weight contributors in the DLR setup is the concrete base which the entire pole is installed on. Naturally, a proper frame is needed to keep the base stable, however, all additional weight needed should be configurable. In Figure L.12 the proposed base design of the portable pole is seen.

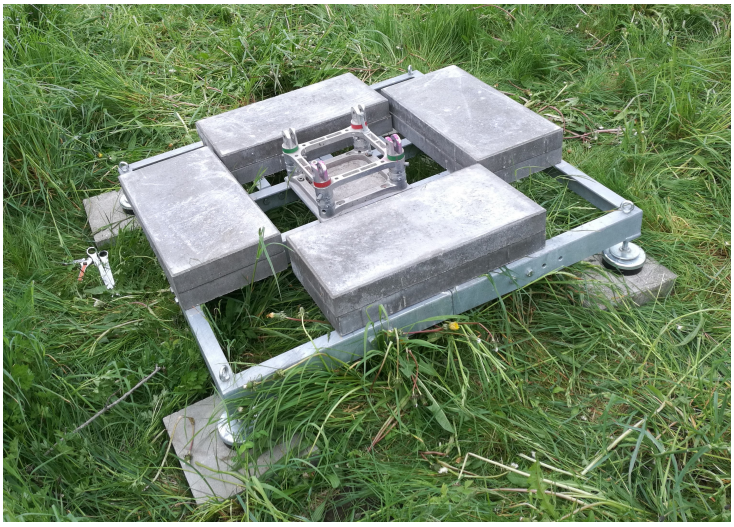


Fig. L.12: The ground base of the portable pole is equipped with tiles, adjustable feet, and a swivel bracket to ease the raising of the lattice mast.

The entire square platform consists of a steel frame containing 4 slots for mounting standard tiles in a vertical rack. The tiles can be acquired in most construction and hardware stores around the world, i.e. 30x60x6cm tiles with a weight of 25kg each. Depending of the required base weight, one of the tiles slot could be used for the equipment cabinet rather than placing it next to the base. Finally, the base platform has 4 adjustable feet for levelling its height on site in case the pavement is not well levelled.

The swivel bracket installed in the middle of the base platform will be used for raising the lattice mast as seen in Figure L.13. The deployment of the portable pole is done by in-stalling tiles in 3 of tiles slots on the base platform leaving 1 slot open. The lattice mast is connected to the swivel bracket in the center of the base platform and put horizontally on the ground in the open tiles slot direction. Our portable pole consists of 5 lattice mast sections, which are 2 meters each providing a 10 meters long lattice mast. The lattice mast and base can be completely separated to ease transportation.

To raise the assembled lattice mast, a steel wire is attached to the mast and

(a) (b)

Fig. L.13: The portable pole can be raised using a swivel bracket installed in the middle of the base platform. The pole is raised using a steel wire connected to a manual winch system.

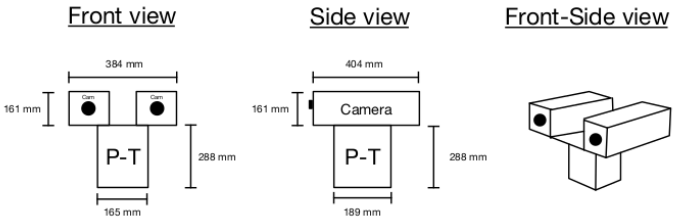


Fig. L.14: The portable pole can be equipped with a camera rig containing two cameras, e.g. RGB and Thermal camera, and a pan-tilt motor to ease view-angle adjustments.

directed towards a temporary installed vertical steel mast on the base platform. On this temporary installed steel mast, a manual winch system is installed, which by the use of hand-power can lift the lattice mast to its operational position where it is locked. Afterwards the temporary equipment is removed, and the last tiles slot is equipped with tiles finalized the deployment of the portable pole. When deployed, 11 tiles are installed in each slot, providing a total weight of 1100 kg in the base framework.

The cameras used to derive the payload for this proposal are defined in the requirements seen in Table L.1. In addition to those cameras, we propose to include the Axis YP3040 Pan-Tilt Motor, as remote camera control has been found desirable for the setup. This, however, increases the minimum required payload weight for the portable pole with 4.2kg.

The Axis YP3040 has a maximum load of 8kg meaning that a custom mounting rig needs to be created to hold both cameras whilst being mounted.

Fig. L.15: The portable pole deployed at traffic intersection. The pan-tilt motor with one RGB camera is installed on the top of the pole, which makes the RGB camera adjustable remotely.

Table L.10: Summary of technical parameters of the TRG-pole.

Operational height [m]	Payload [kg]	Operational base dimensions [L x W x H [m]]	Transport dimensions [L x W x H [m]]	Operational Weight [Kg]	Configurable	Deployment equipment
10	12	1.2 x 1.2 x 10	-	1239	Yes	Van, trailer

The custom mounting rig is seen in Figure L.14. The overall weight of the camera setup, including a buffer, is therefore estimated to be 12 kg.

The final proposal of the TRG-pole in operation mode can be seen in Figure L.15, where you could install your equipment on, e.g. the custom mounting rig. The deployment of the portable pole is 2 hours for 2 persons and requires a van and a trailer. A visual introduction and description of the portable pole can be seen at `https://www.youtube.com/watch?v=SjZlWb3hmBo`. In Table L.10 the specifications of the TRG-pole are summarized.

5 Traffic Analysis using TRG-Pole

The TRG-pole can be deployed in rural areas, which can be of particular use as there in some scenarios are no to limited existing infrastructure (light poles, balconies, trees, etc.) to mount the camera equipment in. For instance, it has been used for a traffic safety analysis as seen in Figure L.16, where there were otherwise limited options besides deploying the TRG-pole.

Fig. L.16: The TRG-pole is deployed at a traffic intersection with limited existing infrastructure.

But what really makes the TRG-pole a great tool, is that the very compact base frame-work allows it to be deployed in most urban areas as well. Though there might exists multiple options in most urban areas, it is however not guaranteed that it provides an ideal capturing angle for the camera equipment. A limited or bad camera view-angle will impact the overall quality of the traffic analysis. An example of this is shown in Figure L.17, where a traffic intersection in Aalborg is used for a traffic analysis study. The left red circle marks a camera mounted in the existing infrastructure, i.e. lighting pole, and the right red circle marks the camera installed in the TRG-pole.

The corresponding output camera feeds are seen in Figure L.18, where the existing infrastructure clearly captures the same objects as the TRG-pole does. The camera installed in the existing infrastructure do however not capture the entire cycling box and the camera's view of field do only allow a limited area of the cycling road after the cyclists begin turning right. Though the TRG-pole is deployed only a few meters away from the lighting pole, the TRG-pole provides a better capturing view for examining the potential conflicts between a cyclist and a right-turning vehicle.

Using semi-automated image processing tools, e.g. RUBA [2], you can use the TRG-pole to conduct traffic analysis with a large variety of scopes, e.g. traffic counts, speed estimations, conflicts, etc. The 10 meters high pole makes a great platform for doing traffic counts as video from such a height is less occlusion prone compared to most existing infrastructure. An example of traffic counts done using the TRG-pole together with RUBA is seen in Figure L.19 , where two detectors were made to register the traffic volumes for respectively one of the entrances to the intersection (A) and one of the

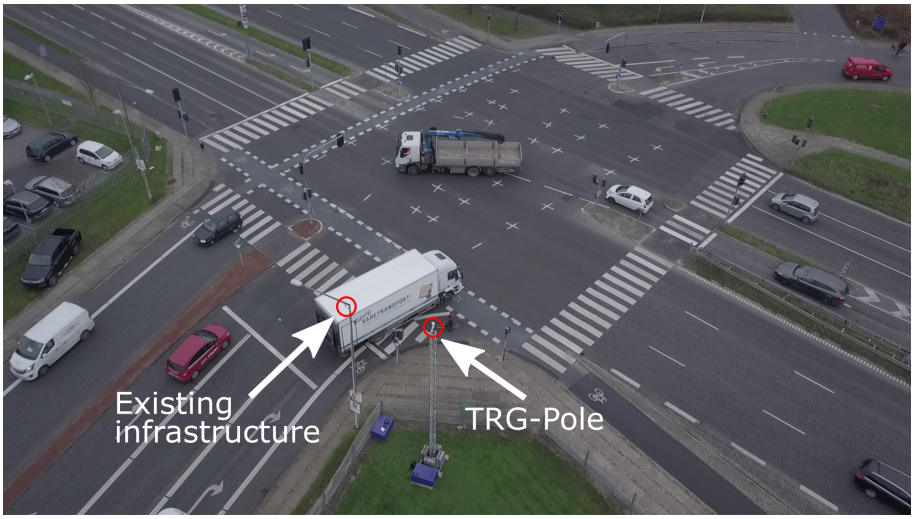


Fig. L.17: The existing infrastructure does not already provide ideal capturing positions for a traffic analysis. The usage of TRG-pole provides more ideal options due to its compactness.

(a) **(b)**

Fig. L.18: Video feed from the camera installed in (A) the existing infrastructure (B) the TRG-pole.

left-turning streams from the main road to the side road (B).

6 Discussion

The presented portable poles types and corresponding solution have been heavily compared and discussed in Section 3.6 in a structured manner given a set of minimum requirements. The requirements have been heuristically derived and the essential requirements defined are thus biased. The remarks made for each of the surveyed solutions is therefore application depended and might still serve beneficial for other application. Most of the type-1 solution, e.g. the Miovision Scout, might be ideal to make a preliminary study at a point

Fig. L.19: Two movement detectors registered the traffic flows through the detectors.

of interest prior to deploying a larger solution. To ensure that the final traffic analysis of the point of interest remains of high quality, the captured data must be of equally good and stable quality. A larger solution is thus necessary to ensure this during longer capturing periods due to various real-life challenges, e.g. weather, vandalism, etc.

The proposed portable pole design is not as easy deployable as most of the type-1 solutions and type-2 solutions, but do not require any guying system for maintaining and ensuring stability. In this proposal it is at most needed as a safety precaution during deployment. The main drawback of the type-4 solution is the transportation weight, which in this hybrid version of type-2 solutions and type-4 solution is reduced while remaining stable during operation. Even though the transportation weight is reduced significantly by removing the tiles from the portable pole base, the frame remains large and made out of steel, meaning that 2 persons and some deployment equipment are still required. An additional drawback of the type-4 solution, and possibly portable poles in general, is the fact they might ruin the naturalistic environment for the drivers, and therefore ruin the desired naturalistic data. The portable pole proposed in this paper do still struggle with this issue, as a portable pole looking similar to the illustration seen in Figure L.15 might still be considered obtrusive in a traffic intersection. But compared to most of the other solution, it is however considered less obtrusive.

The proposed portable pole does to some extent get inspiration and some ideas from the Trivector mobile mast, UTRaCAR, and the DLR platform so-

lutions. These are however all considered to be quite expensive solutions, especially the Trivector mobile mast and UTRaCAR is considered expensive due to the large acquiring and remodeling price of a trailer and a car, respectively. The proposed portable pole is considered a lot cheaper to manufacture due to its simple structure and base framework.

7 Conclusion

This paper presents a survey, proposal, and analysis of portable poles in relation to capturing data in traffic intersection. The surveyed portable pole solutions were split into 4 general types. The type-4 solution appears to fit the defined minimum requirements most, however with a major shortcoming as it is also the lesser mobile and portable pole solution. This leads to the conclusion that for capturing the most stable and useful data, the setup must comprise the lightweight and easy mobility requirements. For the type-1 and type-2 solutions to work, various guying system must be installed on existing infrastructure to fixate the pole. If one involves the existing infrastructure, a better result would be reached if the capturing rig is mounted on the infrastructure rather than using a lightweight or compact portable pole with guying installation. The DLR solution in type-4 is considered to be the best portable pole solution based on vandalism prevention, robustness, stability, and still somehow transportable.

The DLR solution does however not completely fulfill the overall portable pole goal defined in this journal due to the limited mobility. We therefore propose a new portable pole design which combines elements from the type-2 solutions and the type-4 solution so the overall portable pole goal is reached. The proposed portable pole will get the mobility from the type-2 solutions and get the robustness and stability from the type-4 solution. The proposed design is inspired by the type-4 solution from DLR as we also propose to split usage of the portable pole into a transportation stage and an operation stage. The weight of the entire setup can dynamically and with ease be adjusted allowing a more lightweight solution and easier transportation stage. The weight during operation is, however, still intact, such the stability is kept. The proposed portable pole can be deployed by 2 persons in 2 hours in both rural areas as well as urban areas due to its compactness.

Acknowledgements

The authors would like to thank Kay Gimm and Sascha Knake-Langhorst from Deutsches Zentrum für Luft- und Raumfahrt(DLR) for the aid in information about the Test field AIM (Application Platform f or Intelligent Mobility) and

the UTRaCar.
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 635895. This publication reflects only the authors' view. The European Commission is not responsible for any use that may be made of the information it contains.

References

[1] Dlr - institut für verkehrssystemtechnik - utracar und momocar. [Online]. Available: http://www.dlr.de/ts/desktopdefault.aspx/tabid-1237/5441_read-12153/

[2] C. H. Bahnsen, T. K. O. Madsen, M. B. Jensen, H. S. Lahrmann, and T. B. Moeslund, "The ruba watchdog video analysis tool," 2018.

[3] G. Davis and J. Hourdos, "Development of analysis methods using recent data: Shrp2 safety research," *Transportation Research Board of the National Academies, Tech. Rep*, 2012.

[4] European Commission, *Towards a European Road Safety Area: Policy Orientations on Road Safety 2011-2020*. European Commission, 2010. [Online]. Available: https://ec.europa.eu/transport/sites/transport/files/road_safety/pdf/com_20072010_en.pdf

[5] ——, *2017 road safety statistics: What is behind the figures? - Fact Sheet*. European Commission, 2018. [Online]. Available: http://europa.eu/rapid/press-release_MEMO-18-2762_en.pdf

[6] A. Fyhri, H. Sundfør, T. Bjørnskau, and A. Laureshyn, "Safety in numbers for cyclists—conclusions from a multidisciplinary study of seasonal change in interplay and conflicts," *Accident Analysis & Prevention*, vol. 105, pp. 124–133, 2017.

[7] C. Hydén, "The development of a method for traffic safety evaluation: The swedish traffic conflicts technique," *Bulletin Lund Institute of Technology, Department*, no. 70, 1987.

[8] International Transport Forum. Permissible maximum dimensions of lorries in europe. [Online]. Available: https://www.itf-oecd.org/sites/default/files/docs/dimensions_0.pdf

[9] M. Junghans, "Situations- und gefahrenerkennung in verkehrsszenen," in *Kolloquium Verkehrsmanagement und Verkehrstelematik*, Dresden, Germany, May. 8 2013. [Online]. Available: http://www.vimos.org/cms/data/uploads/termine/junghans_sgv.pdf

[10] Litec Strutture & Soluzioni. Flyintower 7.5-500 catalogue. [Online]. Available: http://www.litectruss.com/Litec/media/litec/Downloads/FLYINTOWER_7-5-500_catalogue.pdf?ext=.pdf

[11] T. K. O. Madsen and H. Lahrmann, "Comparison of five bicycle facility designs in signalized intersections using traffic conflict studies," *Transportation research part F: traffic psychology and behaviour*, vol. 46, pp. 438–450, 2017.

[12] C. Masts. Clark masts ft series. [Online]. Available: http://www.clarkmasts.com/media/dyn-docs/products/ft-masts-brochure.pdf

[13] ——. Clark masts model 804/15-6 heavy duty trailer mast. [Online]. Available: http://www.clarkmasts.com/media/dyn-docs/products/model-804-15-6-trailer-mast-cat-no-19877.pdf

[14] ——. Clark masts qt series. [Online]. Available: http://www.clarkmasts.com/products/telescopic-masts/qt-series/

[15] Miovision. Scout video collection unit. [Online]. Available: https://miovision.com/scout/

[16] V. L. Neale, T. A. Dingus, S. G. Klauer, J. Sudweeks, and M. Goodman, "An overview of the 100-car naturalistic study and findings," *National Highway Traffic Safety Administration, Paper*, vol. 5, p. 0400, 2005.

[17] D. I. of Transportation Systems, "Aim mobile traffic acquisition: Instrument toolbox for detection and assessment of traffic behavior," *Journal of large-scale research facilities*, no. 2, A74, 2016.

[18] M. P. Philipsen, M. B. Jensen, R. K. Satzoda, M. M. Trivedi, A. Møgelmose, and T. B. Moeslund, "Day and night-time drive analysis using stereo vision for naturalistic driving studies," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, June 2015, pp. 1226–1231.

[19] S. Plainis, I. Murray, and I. Pallikaris, "Road traffic casualties: understanding the night-time death toll," *Injury Prevention*, vol. 12, no. 2, pp. 125–138, 2006.

[20] L. Sakshaug, A. Laureshyn, Å. Svensson, and C. Hydén, "Cyclists in roundabouts—different design solutions," *Accident Analysis & Prevention*, vol. 42, no. 4, pp. 1338–1351, 2010.

[21] Transport-, Bygnings- og Boligministeriet. Bekendtgørelse om køretøjers største bredde, længde, højde, vægt og akseltryk (dimensionsbekendtgørelsen). [Online]. Available: https://www.retsinformation.dk/pdfPrint.aspx?id=137554

Paper M

The RUBA Watchdog Video Analysis Tool

Chris H. Bahnsen, Tanja K. O. Madsen, Morten Bornø Jensen,
Harry Lahrmann, and Thomas B Moeslund

This article has been submitted as a public deliverable within the
InDeV EU project, 2018. Based on the public wiki page at
`https://bitbucket.org/aauvap/ruba/wiki/`.

The layout has been revised.

Fig. M.1: The RUBA logo.

1 Introduction

The Road User Behaviour Analysis (RUBA) project is a watch-dog tool for computer-based analysis of traffic videos. The program can be used on Windows, MacOS, and Linux computers.

RUBA is developed by the Visual Analysis of People Lab at Aalborg University, Denmark, in collaboration with the Traffic Safety Research Group at Aalborg University.

RUBA allows the user to draw fields (detectors) on the video image by using a simple click-based drawing tool. The sensitivity of the detector, regarding movement in the image, is adjusted by different parameters in the program.

How to contribute

Please feel free to use RUBA and see if it fits your use case and research needs. If you encounter a bug by doing so, or if you have any suggestions on the further improvement of RUBA, please report it in our issue tracker.

License

2 Analysis in RUBA

The procedure when conducting an analysis in RUBA is as follows:

1. Import video(s)
2. Create module(s) for the analysis
3. Calibrate parameters to ensure that the right movements/road users are registered
4. Run the analysis

2.1 Import of videos

RUBA handles videos of most file types and resolutions. The program offers two different approaches for handling the synchronisation and time management for every frame of a video file:

1. If the frame rate of the video is constant, the start time of the video might be encoded into the file name of the video. The exact time of a frame will be computed based on the start time, the frame rate of the video, and the current frame number.

2. If the frame rate of the video is varying, you may put the exact date and time of each frame in a separate log file. The log file should be placed in the same directory as the corresponding video file and share the same file name except for the extension. As default, RUBA looks for corresponding files with the '.log' extension.

For more information on the video synchronisation options, refer to Section 4.2. Once you have selected a suitable way to ensure the synchronisation of the video, you have two options to import video files into RUBA, illustrated in Figure M.2 and listed below:

1. Use File -> Load Video Files or click the button at the menu bar (CTRL + O). This option will clear the current list of video files and import the new files that you have selected.

2. Use the 'Add videos to list (CTRL + INS)' button in the 'Video files' pane. This option will add the selected videos files to the bottom of the current list of video files.

2.2 Creation of modules for the analysis

After the videos have been imported the first video is shown in the window pane. A module for the analysis is created by pressing the button for the desired module. This is illustrated in Figure M.3.

Choose the desired detector type as illustrated in Figure M.4. Then press OK. A description of the different detectors is given in Section 5.

2.3 Drawing the mask

After the desired detector have been chosen a new window opens, shown in Figure M.5. This window contains the settings of the detector and lets the user draw the detector. Via `Configure detectors` the detector is chosen, after which drawing tools to create the detector and a number of detector settings appears. The settings depend on the chosen detector type.

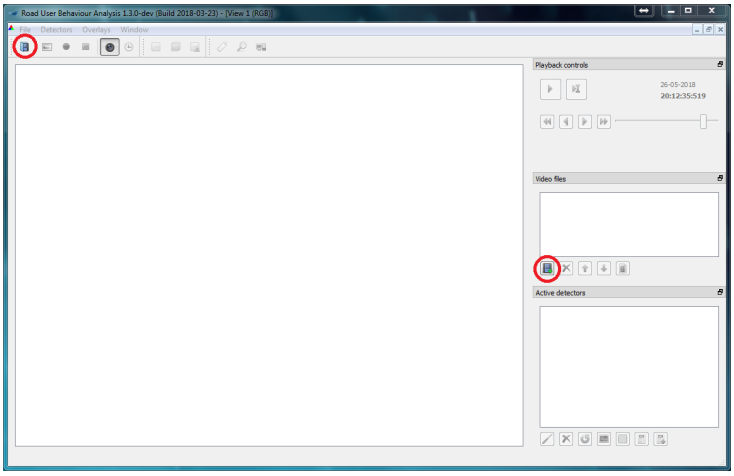


Fig. M.2: Import of videos is done via either of the two buttons marked in red.

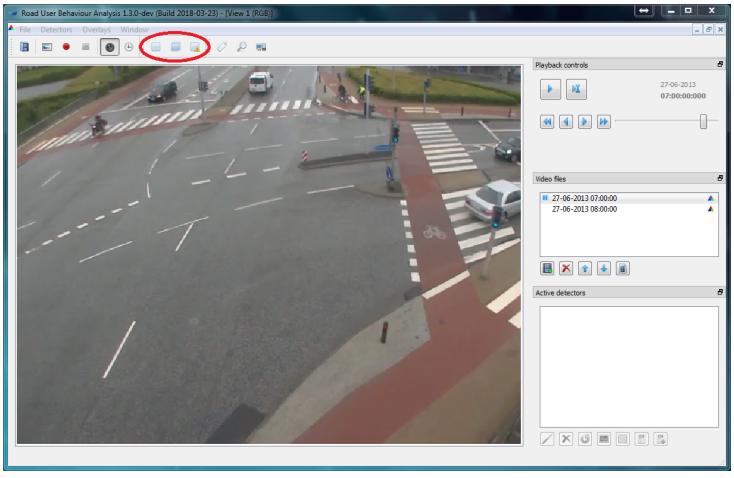


Fig. M.3: Creation of modules

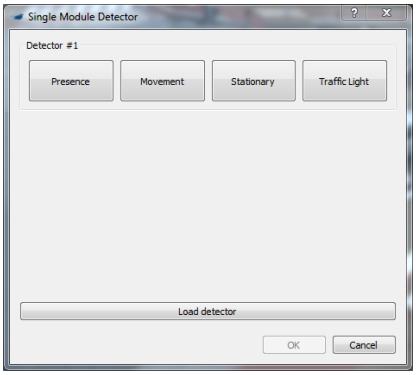


Fig. M.4: Choice of detector type in single module

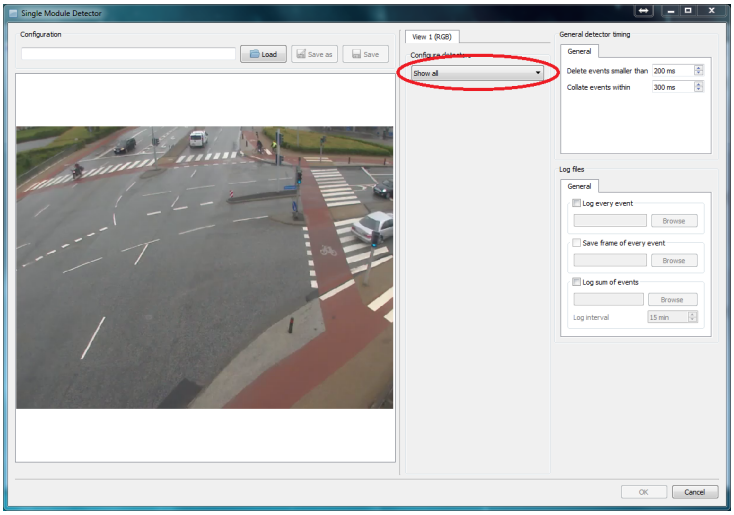
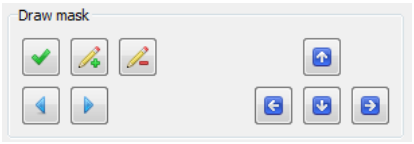
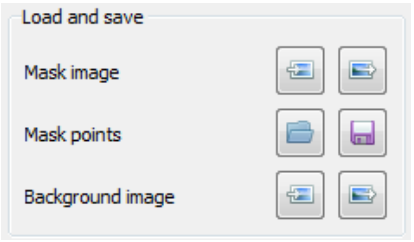


Fig. M.5: Creation of detectors.



(a) Tools for drawing the mask



(b) Load an existing detector mask or create images and detector masks to reuse the masks

Fig. M.6: Mask manipulation tools in RUBA.

Table M.1: Functionality of the drawing tools in RUBA.

Button	Description
	Remove the last corner point.
	Add a point between the current and the previous corner.
	Switch between corners. The current point is marked with a circle.
	Move the corner up/down/left/right.

To draw the outline of the detector, click on the pencil. The detector is drawn by clicking in the image. Straight lines are created between the points. The latest point can be deleted by right clicking.

Drawing tips

Use the drawing tools illustrated in Figure M.6a to modify your detector. The functionality of the tools is explained in Table M.1.

You can use keyboard shortcuts in RUBA. Hold your mouse over the buttons to find the keyboard shortcut.

Move points: When drawing your detector, you can click on the points and drag them to where you want them to be.

If you have an existing detector mask or want to save the mask area as an image or RUBA file, you can use the functionalities in the `Load and save` panel shown in Figure M.6b.

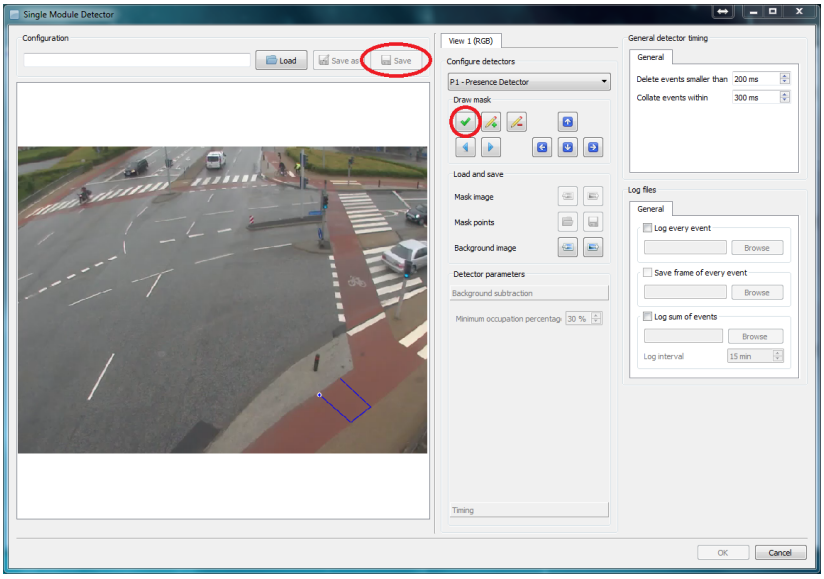


Fig. M.7: Creation of detector using the drawing tool

- `Mask image` loads (left) or saves (right) an image of the detector where the detector is white and the background is black.
- `Mask points` can be used to save a RUBA configuration file with information on the size and position of the detector.
- `Background image` imports and exports a screen shot of the background.

Once the detector has been drawn (i.e. it only needs to be closed), double click or press the green tick mark, marked in red on Figure M.7. After this, the parameters can be adjusted, and it can be chosen if logs should be created. See Section 7.3 for detailed information on the log system of RUBA.

The detector is saved via the `Save`-button before the window is closed via the `OK`-button. Configuration files that have previously been saved can similarly be imported in this window.

2.4 Calibration of parameters

To ensure that the right objects are detected the parameters must be calibrated. This is done via a number of tools, marked in red in Figure M.8, which let the user gain insight into what is detected by the algorithms.

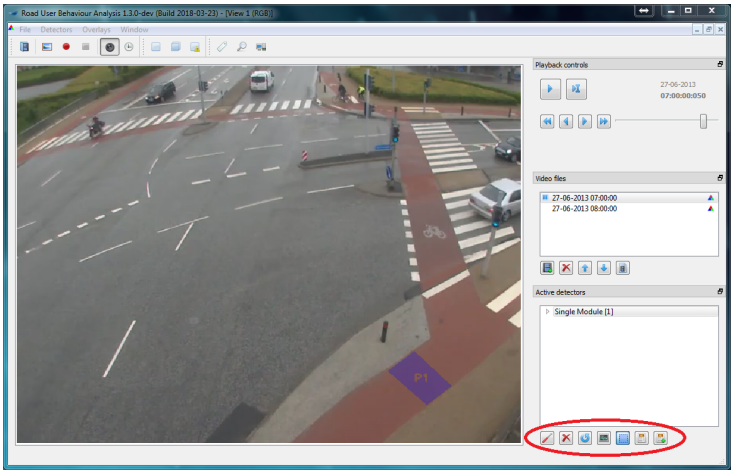


Fig. M.8: Tools for calibration of the detectors. From left: 1) edit the detector. Editing can also be done by right clicking on the detector in the `Active detectors`; window. 2) delete the marked detector. If the detector has not been saved, it is deleted completely and cannot be imported. 3) histograms. 4) Overlay of the detector in the image. 5) information about activity in the detector. 6) extended information about the detector.

2.5 Histograms

The most important tool for the calibration is the histograms of the activity in each detector.

Histograms are used to adjust the detectors. If the amount of activity is sufficiently high so that the software recognizes it as a road user, the activity is marked with a bright colour, illustrated in Figure M.9. If the parameters of the detector are not adjusted correctly, then the road user will either be missed or only partly detected, as seen in Figure M.9a. After the adjustments of the parameters the road user will be clearly detected.

The histograms of the movement detector, the stationary detector, and the traffic light detector are described in more details in Section 5.

When adjusting the detectors, change a few parameters at a time and validate experimentally if the change has any effect. The most important parameters of the detectors are listed below:

- **Presence detector:** Minimum occupation percentage
- **Movement detector:** Trigger threshold, movement range, and minimum speed
- **Stationary detector:** Minimum occupation percentage, minimum speed, and max vector count
- **Traffic light detector:** The position of the annotated traffic light positions

(a) Before calibration **(b)** After calibration

Fig. M.9: Calibrating the trigger threshold of the presence detector.

A detailed description of all detector parameters is given in Section 5. Detailed information on how to set up the logger is given in Section 7.3. There, you will also find information on the timing options of the different detector modules.

2.6 Run the analysis

When the detectors have been calibrated the analysis can be performed. If it has not yet been specified which log files should be created during the analysis, this is done by double clicking the detector in *active detectors*. Run the analysis by pressing the `play` button which is marked in red on Figure M.10.

2.7 Inspecting the log files

If `log every event` or `log sum of events` have been marked when configuring the detector, a number of log files (.csv-files) will be generated. The log files may be inspected by a text editor or a spreadsheet program such as Microsoft Excel. More details on the log system are provided in Section 7.3.

2.8 Multi-threaded processing

It takes time to process long videos in RUBA, especially if the resolution of the video frames is high. In order to help with this problem, RUBA has an option to split the analysis such that it runs on multiple threads. Once the videos are

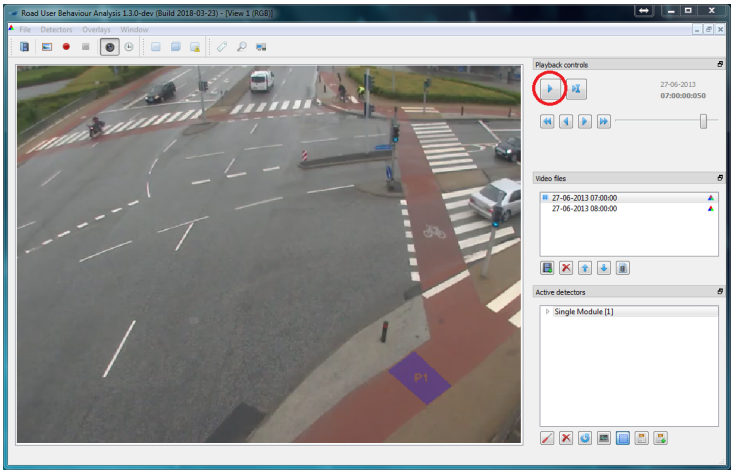


Fig. M.10: Running a traffic analysis. Double click on the first video to scroll back to the beginning of the video, and then press the play button to run the analysis. The analysis is complete when the time stops in the end of the last video. Please note, that the time and date must be specified in the beginning of each video if the user defined time format is used.

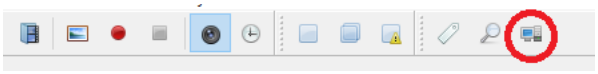


Fig. M.11: Opening the Multi-threaded processing dialogue.

loaded and the detectors are initialised, press the `Perform multi-threaded processing` button in the main menu, marked in red in Figure M.11.

Once you have opened the multi-threaded processing dialogue, you may select the desired number of threads to perform the analysis. The maximum number of threads is dependent on the number of physical CPU-cores on your machine. Furthermore, in order to create a number of threads, each tread needs at least one unique video.

In the example in Figure M.12, RUBA has detected that the machine has eight CPU-cores, but RUBA only allows the creation of four threads because only four video files are loaded. In order to increase the number of threads, more video files should be provided and the multi-threaded processing dialogue should be reopened.

The optimal number of processing threads

The maximum number of threads is computed as the number of CPU-cores, minus 1. If the computer has four CPU-cores, three will be selected for running the analysis - and the last will be spared for showing the progress in RUBA and for running other tasks.

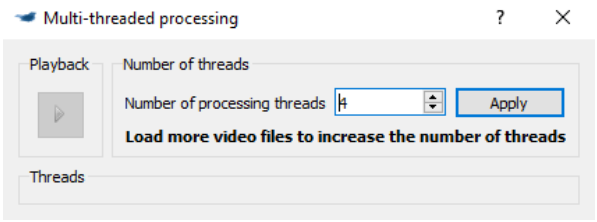


Fig. M.12: The initial window in the Multi-threaded processing dialogue.

As a general rule, the processing speed is proportional to the number of processing threads. However, if your CPU features hyper-threading technology, RUBA will typically see one physical CPU-core as two CPU-cores. On a machine that has four physical CPU-cores with hyper-threading, RUBA will report the maximum number of threads to `4 * 2 - 1 = 7`, seven threads. In this case, the addition of more threads than physical CPU-cores will have little impact on performance.

Running the multi-threaded analysis

Once you have selected the desired number of processing threads, press the `Apply` button. Behind the scenes, RUBA will save the detectors and reload them for every thread. This might take some seconds depending on the number of threads and the size of the detectors. After this process has finished, the window will be resized and the desired number of threads are shown. A sample screen with four threads is shown in Figure M.13. Press the `Play` button in the upper left corner to start processing.

The `Video Files` window shows the progress of the analysis. As opposed to normal analysis, it is not possible to jump to a specific video by double-clicking.

The `Detectors` window allows you to inspect the progress by expanding the arrows, similar to a normal analysis. However, the following features are not supported when the multi-processing window is opened:

1. Reconfiguring detectors
2. Showing the detector masks
3. Showing the detector histograms
4. Overlaying debug information on the videos

Because the analysis now runs in parallel, you will find that RUBA creates temporary log files, one for each thread. Once all the threads has finished processing, RUBA will automatically combine the temporary log files into a single log file.

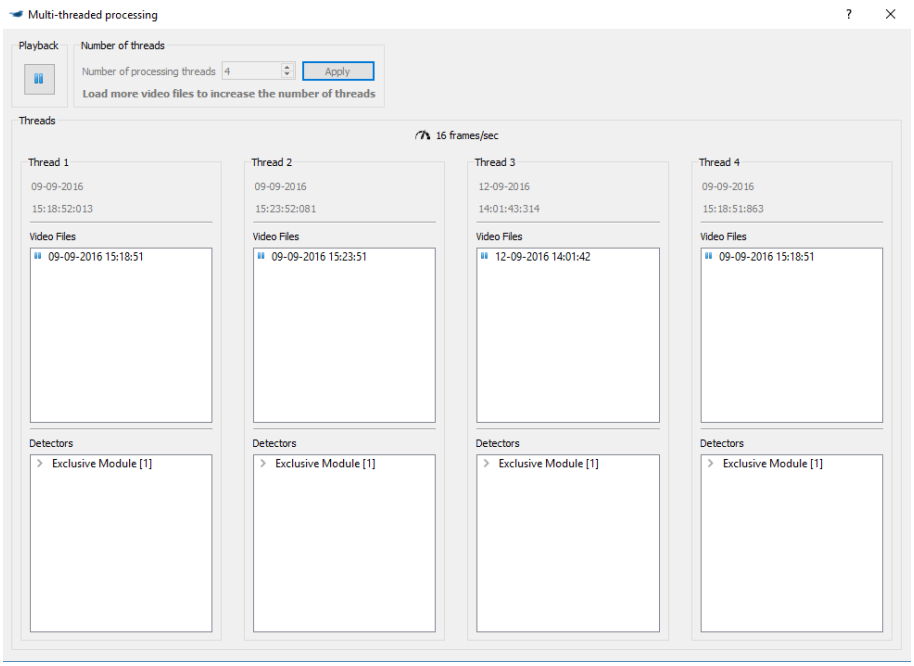


Fig. M.13: The multi-threaded processing is started.

3 User interface

Figure M.14 illustrates the main window of RUBA, after a video has been imported. Until then, most buttons are inactive. In the following, the function of each button is described. Keyboard short cuts are defined in square brackets.

1. Main menu. In the tabs, the same functions that are accessible via buttons in the main window can be found, as well as a number of program settings to use before conducting the analysis.

2. Import video(s) [Ctrl + O].

3. Take a screenshot of the video pane (20) [F9].

4. Record a video of the video pane. Clicking the red dot [F10] starts the recording. It is possible to pause the recording by clicking the button again. The recording can be resumed by clicking the red button again. When clicking the square button [ctrl + F10] the recording is finished. All overlays (detectors, etc.) which are shown in the video pane (20) will appear in the recording.

5. Enable/disable that the video is shown in the video pane (20) [F3].

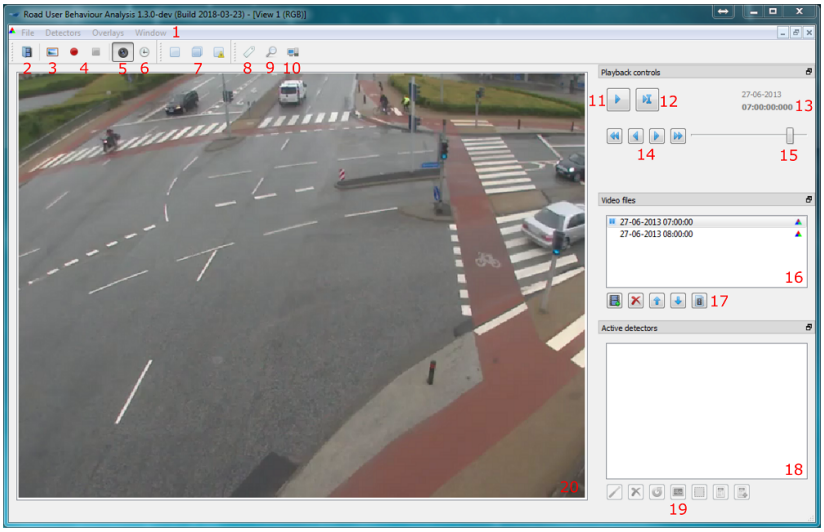


Fig. M.14: User interface shown when one or several videos have been imported.

6. Add/remove overlay which shows a timestamp (13) in the video [F4].

7. Flexible analysis tool (detector) consisting of a single module (left) [Ctrl+1], a double module (middle) [Ctrl+2] and an exclusive module (right) [ctrl+2].

8. Annotate Ground Truth. Opens the `Ground Truth Annotator` panel which can be used to manually detect activity. These detections can be used to calibrate detectors.

9. Review Log Files. Opens the `Log File Reviewer` panel which can be used to review events from a log file and create a new log file with selected events.

10. Multi-Threaded Processing. Press this button to open the `Multi-Threaded Processing` panel and analyse the videos in multiple thread simultaneously to speed up the analysis.

11. Start/pause analysis [space].

12. Jump to a specific frame in the video.

13. Date and time for the video.

14. Navigate through the video. Use these four buttons to respectively jump five frames previous [A], one frame previous [S], one frame forward [D] and five frames forward [F].

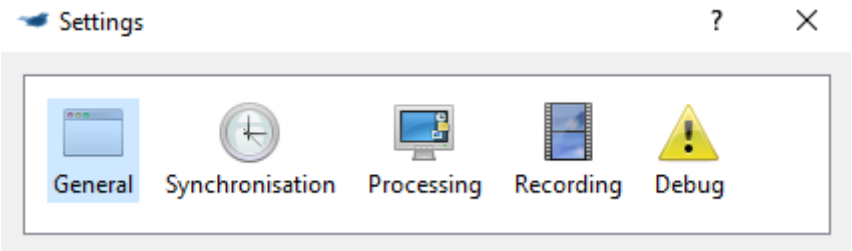


Fig. M.15: The settings panel in RUBA. The `Debug` panel is only shown when running a development version of RUBA.

15. Adjust video speed. When the slider is placed to the far left the video is paused. When the slider is placed to the far right the video is sped to the maximum.

16. Imported videos. The video that is currently played is marked with a *pause* symbol.

17. Respectively add videos [Ctrl+insert], delete imported videos [Ctrl+del], change the order of the videos [ctrl+arrow up] and [ctrl+arrow down] and show the video properties. The button to the far right contains properties for the video (start/end time, frame rate, file name and resolution).

18. Inserted/created detectors.

19. Support tools for creating and calibrating of detectors. From left: 1) edit the detector [ctrl+R]. 2) delete the marked detector [Del]. 3) reset the detector. 4) histograms [F5]. 5) Overlay of the detector in the image [F6]. 6) information about activity in the detector [F7]. 7) extended information about the detector [F8].

20. Video pane.

4 Settings

Access the settings under File -> Settings and a settings panel similar to Figure M.15 will be shown.

4.1 General

The general settings pane is shown in Figure M.16.

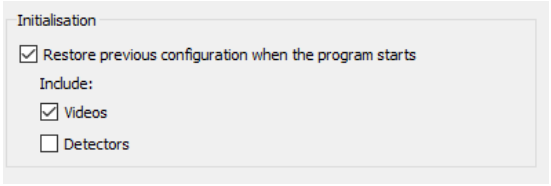


Fig. M.16: The general settings in RUBA.

- **Restore previous configuration when the program starts:** When you open RUBA it will try to load the videos that you imported the last time you were running RUBA. You can change this behaviour or choose to also load the detectors from the last time you were running RUBA.
- An alternative is to use the `Load configuration` and `Save current configuration` buttons in the `File` menu. A configuration file is, similar to the detector files, an .yml-file, but contains references to the videos and detectors that are currently loaded into RUBA. By using this functionality, you can quickly switch between different combinations of videos and detectors.

4.2 Video synchronisation

The video synchronization pane is shown in Figure M.16.

- **Start time of each video is encoded in the file name:** If the frame rate of the video is constant, the start time of the video might be encoded into the file name of the video. The exact time of a frame will be computed based on the start time, the frame rate of the video, and the current frame number.
  - **Frame rate:** Used to define the frame rate. The value should match the frame rate of which the video is recorded. It is recommended to let RUBA auto-detect the frame rate (default).
  - **Date and time:** Used to define the date and time of which the video is recorded. This can be encoded in the file name, so that the information can be imported automatically. The format is chosen as either:
    - `MM-dd-HH` (month-day-hour). The year must be specified manually when playing the video.
    - `yyyy-MM-dd` (year-month-day)
    - `yyyyMMdd-HH` (year-month-day-hour)
    - `yyyy-MM-dd-HH` (year-month-day-hour)

    * `yyyyMMdd-HH-mm-ss` (year month day-hour-minute-second)
    * `yyyyMMdd-HH-mm-ss.zzz`
      (year month day-hour-minute-second.millisecond)
    * `user defined` in which the date and time is specified manually every time the video is played from the beginning.

- **Time stamps of each video frame are provided in a separate log file:** If the frame rate of the video is varying, you may put the exact date and time of each frame in a separate log file. The log file should be placed in the same directory as the corresponding video file and share the same file name except for the extension.
  - As default, RUBA looks for corresponding files with the '.log' extension. You may change this if necessary.
  - Each line of the log file should be contain the frame number and the frame time in the following format: `frameNbr yyyy MM dd hh:mm:ss.zzz`. An example is shown in Figure M.17c.

4.3 Processing

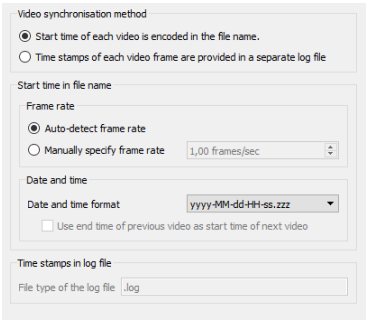
The video processing pane is shown in Figure M.17b.

- **Playback speed:** Used to decide the speed of which the video will be analysed. The speed can be altered later on.
- **Skip frames:** In order to speed up processing, RUBA may skip every n'th frame. Beware, however, that this may affect the accuracy of the detectors.
- **Resolution:** Used to create a warning if the imported video is recorded at a low resolution. The width and height of when the warning is created can be set manually.

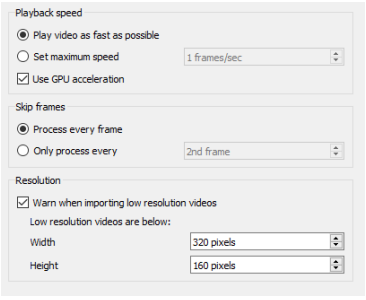
4.4 Recording

The behaviour of the built-in video recording may be altered from the settings pane shown in Figure M.17d. As default, RUBA records to a single file when the `Record video to file` button is pressed. However, when log files are played back with the Log File Reviewer, it might be beneficial to create a separate recording for each event.

If the option `Create a new recording for each jump of at least` is selected, a new recording will be created when the `jump to frame` functionality is used, either directly or indirectly from the Log File Reviewer. If this option is unchecked, a single video file will be created that contains the same 'jumps' as the original playback from RUBA.



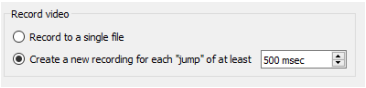
(a) The synchronisation settings in RUBA.



(b) The processing settings in RUBA.

```
00000 2016 08 25 12:02:16.215
00001 2016 08 25 12:02:16.248
00002 2016 08 25 12:02:16.282
00003 2016 08 25 12:02:16.315
00004 2016 08 25 12:02:16.348
00005 2016 08 25 12:02:16.382
00006 2016 08 25 12:02:16.415
00007 2016 08 25 12:02:16.448
00008 2016 08 25 12:02:16.481
00009 2016 08 25 12:02:16.514
00010 2016 08 25 12:02:16.548
00011 2016 08 25 12:02:16.581
```

(c) A sample log file containing time stamps for every individual frame.



(d) The recording settings in RUBA.

Fig. M.17: Settings panels in RUBA.

Table M.2: Applications of the detector types in RUBA.

Detector	Description
Presence	Simple analysis and traffic counts. Traffic counts in sections. NB! The highest accuracy is obtained if the traffic streams are separated.
Movement	Analysis and traffic counts in areas shared by road users from different directions (e.g. in intersections). Road users driving in the opposite direction of travel.
Stationary	Analysis of road users that do not move. Detection of parked cars.
Traffic light	Analysis of the phases of one or several traffic lights. Combined with other detectors, analysis of red light running.

Table M.3: Adjustable parameters in the detector modules.

	Presence	Movement	Stationary
Trigger threshold		x	
Minimum occupation percentage	x		x
Minimum speed		x	x
Movement direction		x	
Max vector count			x
Max triggered duration	x	x	x

5 Detector Types

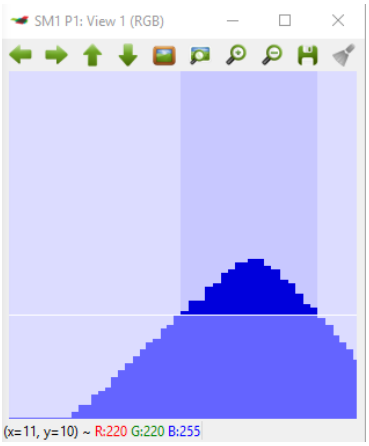
The algorithm behind the software consists of four detector types (presence, movement, stationary, and traffic light) with different attributes. Examples of the application of the four detector types are shown in Table M.2.

Combinations of the detectors are introduced on the description of the detector modules in Section 6.

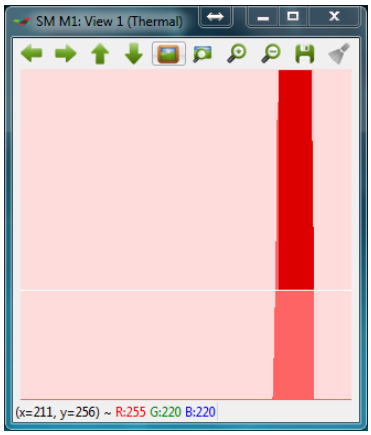
An overview of the adjustable parameters for each detector type is given in Table M.3. A detailed description of the parameters is given with the overall description of each detector type.

5.1 Presence Detector

The *presence detector* checks if there is an object in a specific area of the video. With this method objects (vehicles, road users) that are not a part of the background (the road, the surroundings) are extracted. This is done by converting the image to a gray scale image and finding presences (continuous



(a) Histogram of the presence detector. X-axis: time Y-axis: registered occupation percentage, in the range from 0 to 100.



(b) Movement detector histogram. X-axis: time. Y-axis: activity through the detector (the higher red lines, the more activity was registered

Fig. M.18: Histograms of the movement and presence detectors.

lines), where large variations in the contrast appear. In this way the algorithm finds road markings, changes in the pavement, and road users. This is done for all the frames of the video. For two consecutive frames, vectors between the lines are created and summed up. In this way we get a measure for the activity which is based partly on the size of the object, partly on the speed of the object moving across the area. In order to take noise in the image into consideration; i.e. from small changes in the contract, birds, movement of leaves, or shadows, the sensitivity of the presence detector is controlled by some parameters. The background is updated regularly to extract elements that are consistent in the image for a long time, or elements that occur due to changes in the light conditions and the creation of shadows.

Presence detector histogram

The *lower white, horizontal line* of the presence detector histogram of Figure M.18a shows if the size of the object that is present inside the detector is lower or higher than the **Minimum occupation percentage**.

To be detected as a road user, the blue lines must go higher than the horizontal line and the width of the blue lines above the threshold must be above the time interval which is defined in **Delete events smaller than** (standard setting: 200 ms). Only the dark blue lines counts in the time that it should be above the threshold to be detected as a road user. The width of the dark blue part should be approx. 0.5 cm when using the standard setting

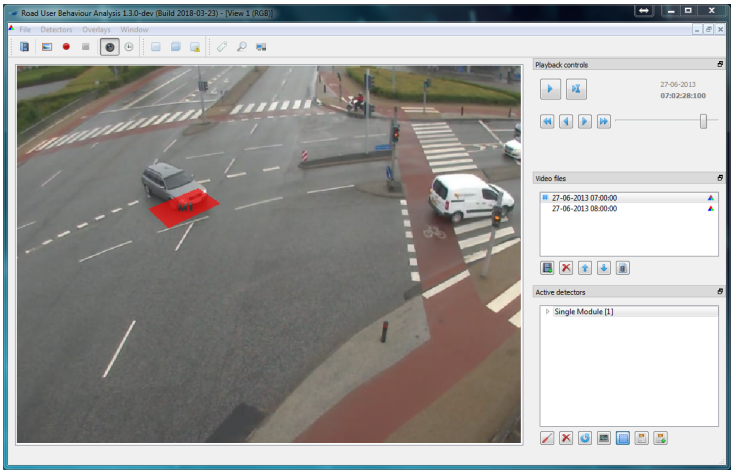


Fig. M.19: The movement detector (red). The movement detector detects activity (movement) in a specific direction through the detector.

of 200 ms. If the activity for one road user results in two tops with a small gap in between (meaning that it will be registered twice), you can adjust the value for **Collate events within** (standard setting: 300 ms). In this way you can influence how large the gap between two tops can be before they should be registered as two separate road users. NB! Be careful if changing this value. If too big, two cars with a small gap between them will be registered as one.

Presence detector parameters

Minimum occupation percentage Fraction of the defined mask that must be occupied by a road user or a temporary object in the scene. Use this value to filter out noise or small road users, for instance pedestrians and bicyclists.

5.2 Movement Detector

The *movement detector* checks if there is activity in a specific direction in a certain area of the video by means of the *Farnebäck dense optical movement estimation*(Farnebäck, 2003). In the movement detector points in two consecutive frames are identified and matched. Objects moving across the detector will result in vectors that are dependent on the direction and speed of the object. Higher speeds will result in larger vectors. The amount of vectors per direction (in terms of degrees) is summed up for those vectors that are higher than a predefined value of the speed of the object. Vectors below this value are omitted.

Movement detector histogram

The histogram of the movement detector is illustrated in Figure M.18b. The *red, vertical lines* of the histogram indicate that something has moved faster than the **Minimum speed** in the specified direction. The unit is not comparable to known speed units (e.g. m/s or km/h) but low values means that slow road users will be detected, high values that only fast objects will be detected. Choosing a narrow range of direction will result in limited activity and hence few/short red lines.

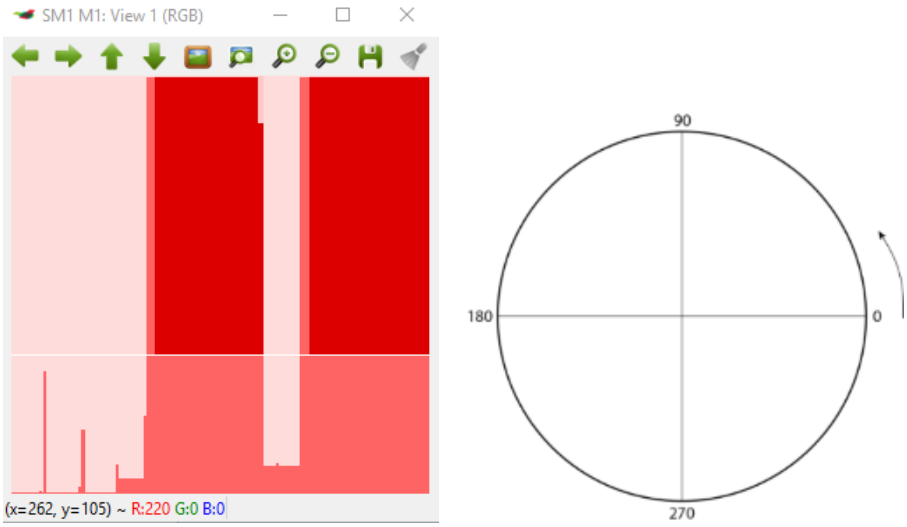
The *white, horizontal line* shows the Trigger threshold, i.e. how much activity is required to be identified as a road user (above the line) and what is considered as noise (below the line). To be detected as a road user, the red lines must go higher than the horizontal line and the width of the red lines above the threshold must be above the time interval which is defined in **Delete events smaller than** (standard setting: 200 ms). Only the dark red lines counts in the time that it should be above the threshold to be detected as a road user. The width of the dark red part should be approx. 0.5 cm when using the standard setting of 200 ms. If the activity for one road user results in two tops with a small gap in between (meaning that it will be registered twice), you can adjust the value for **Collate events within** (standard setting: 300 ms). In this way you can influence how large the gap between two tops can be before they should be registered as two separate road users. NB! Be careful if changing this value. If too big, two cars with a small gap between them will be registered as one.

Movement detector parameters

Trigger threshold Limit for when an activity will be registered. The parameter is used to sort out noise in the video. In the histogram of Figure M.20a, the trigger threshold is shown as a horizontal line. To filter out noise, the red line must be above the horizontal line in four out of the last ten frames. This is visible from the histogram below; the red line is above the horizontal line in a short duration (three frames) before the detector is finally triggered and turns dark red.

Minimum speed Measure for how fast an object must move to be registered in the movement detector. The higher value, the faster it has to move. The minimum speed is measured in pixels.

Flow range Defines in which direction the vectors must go if the activity should be registered as an event. The range can be chosen on a circle (0-360 degrees), illustrated in Figure M.20b. The range is chosen by either inserting the range in the fields or by dragging in the dots on the circles.



(a) Trigger threshold **(b)** Flow range

Fig. M.20: Histogram and flow range of the movement detector.

5.3 Stationary Detector

The *stationary detector*, depicted in Figure M.21, detects if an object is idling or moves very slowly through a specific area of the video by means of a combination of the presence and movement detectors. To detect an object (a road user) the presence detector must be triggered, while the movement detector must not be triggered. This indicates that an object is present in the area but is moving very slowly or not moving at all.

Stationary detector histogram

The histogram of the stationary detector, shown in Figure M.22, is a bit different than the histograms of the other detectors, as the detector is a combination of the presence (blue) and movement (red) detectors.

The *lower white, horizontal line* of the histogram shows if the size of the object that is present inside the detector is lower or higher than the **Minimum occupation percentage**.

The *upper white, horizontal line* shows the limit of how much the object can move and still be registered as standing still. The height of the red line shows the amount of activity in any direction with a speed higher than the **Minimum speed**. To be detected as a road user standing still, the blue lines must reach the lower white, horizontal line and the red lines must not exceed the upper white, horizontal line. If both of these criteria are met, the blue

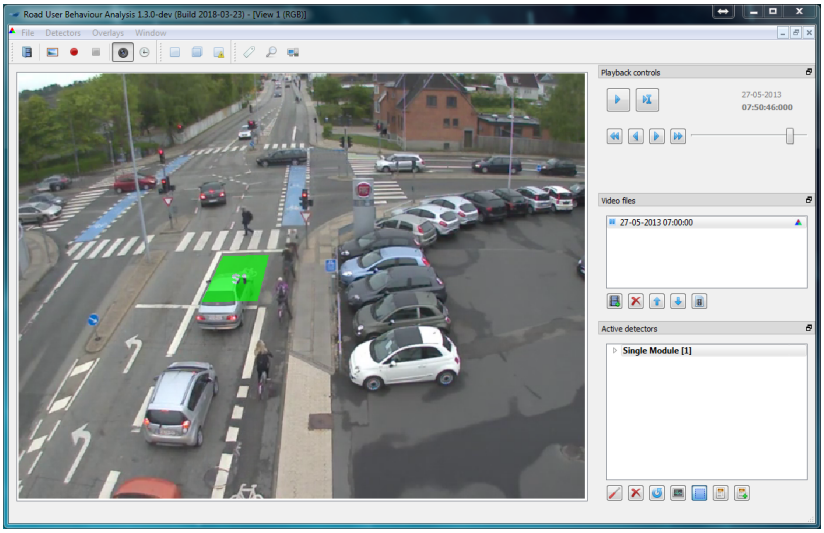


Fig. M.21: The stationary detector (green). The stationary detector detects if something is idling or moves slowly through the area covered by the detector

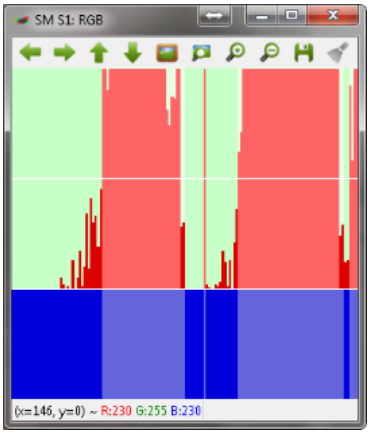


Fig. M.22: Stationary detector histogram. X-axis: time Y-axis: activity through the detector (the higher lines, the more activity was registered))

and red lines will turn into a brighter colour. If the width of bright coloured lines is above the time interval which is defined in **Delete events smaller than** (standard setting: 200 ms), a road user is detected. The width of the dark red part should be approx. 0.5 cm when using the standard setting of 200 ms. If there is a small gaps in the bright coloured lines, it will still be registered as one road user if the gap is smaller than the value for **Collate events within** (standard setting: 300 ms). NB! Be careful if changing this value.

Stationary detector parameters

Minimum occupation percentage See the description of the equivalent parameter for the presence detector.

Minimum speed See the description of the equivalent parameter for the movement detector.

Max vector count The maximum amount of vectors that is allowed to be above the defined minimum speed to result in an event. In other words; the maximum allowed amount of 'movement' in the mask. If the movement is larger than this, we do not consider the mask to be stationary.

5.4 Traffic Light Detector

The *traffic light detector*, illustrated in Figure M.23, detects the different phases (red, yellow, red-yellow, and green) of a traffic light. The detector mask is to be defined in a small region around the traffic light and is used to perform image stabilisation. Image stabilisation is performed in order to make sure that the annotated traffic light positions follows the actual positions of the traffic light in case of small movements (oscillations) of the camera.

The traffic light positions should be annotated in the centre of the traffic light. An overview of the detected states of a traffic light is given in Table M.4.

Traffic light detector histogram

The histogram of the traffic light detector visualises the detected state of the traffic light. Two examples are shown in Figure M.24.

The five states of the traffic light are displayed in different colours that may be seen from the annotated histogram of Figure M.24b.

6 Detector Modules

The four detectors can be combined in detector modules. The modules manage logic between one or two detectors. Furthermore, the modules define when a

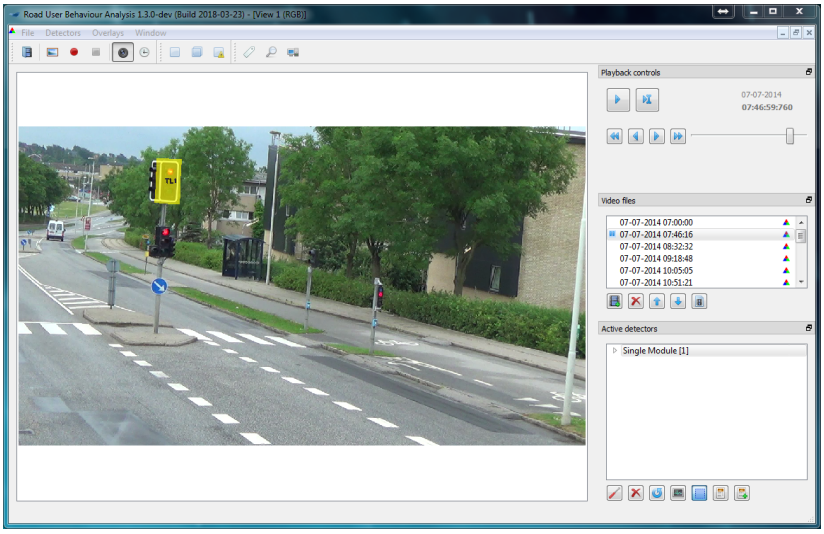
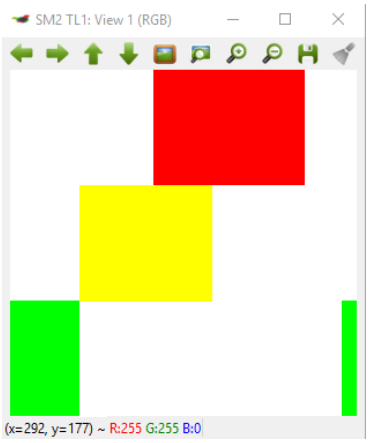


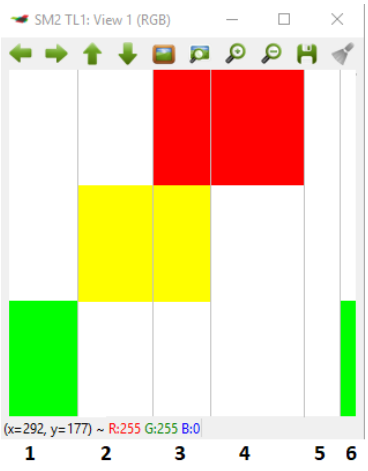
Fig. M.23: The traffic light detector (yellow). The traffic light detector detects the colour of the traffic signal

Table M.4: Overview of the possible states of the traffic light and the corresponding detections as defined by the colour trigger. The ambiguous state is activated if the detector is unsure of the state of the traffic light.

Traffic light state	Colour trigger		
	Red	**Yellow**	**Green**
Red	x		
Red-yellow	x	x	
Yellow		x	
Green			x
Ambiguous			



(a) X-axis: time. Y-axis: detected state of the traffic light



(b) Detected states in order of appearance (left to right): (1) Green. (2) Yellow. (3) Red-yellow. (4) Red. (5) Ambiguous (blank). (6) Green

Fig. M.24: Histograms of the traffic light detector.

detector takes on one of three states:

1. *Activated:* When a detector is activated, movement in the field can be registered.
2. *Triggered:* The detector has registered activity of the right type (e.g. the right direction) and in an extent that indicates that the movement comes from a road user (and not just noise in the image).
3. *Flagged* (results in the detection of an event) When the detector has been triggered for a number of consecutive frames, an event is registered and saved in a log file.

RUBA consists of a single module (one detector), a double module (two detectors), and an exclusive module (two detectors).

6.1 Single

The single module consists of one detector type (presence/movement/stationary-/traffic light). An event is saved in a log file when the criteria are met according to the specifications of the chosen detector. An example is shown in Figure M.25.

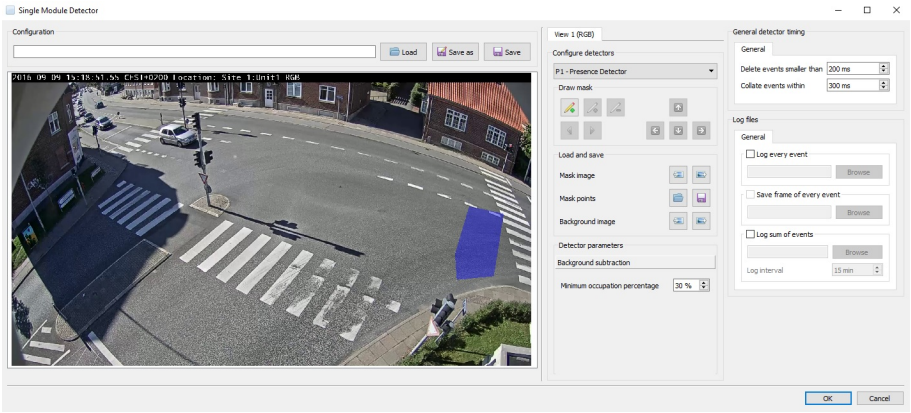


Fig. M.25: Example of a single module. The single module lets the user create one detector of one's own choice per module.

6.2 Double

Consists of two optional detectors. An event is detected and saved in the log file when both detectors have been triggered within a specific time distance defined by the user. An example is shown in Figure M.26.

The timewise relation between the two individual detectors can be defined in two ways. *Interval timing* can be used to define the maximum time gap between the two detectors are triggered or stop being triggered. For instance, the time can be defined as the the interval from a vehicle enters one detector and another vehicle enters the other detector. *Overlap timing* is used when the two detectors should be activated simultaneously. It is possible to define a buffer so that events can be registered if the detectors are activated almost at the same time. Detailed information on the timing options is given in Section 7.3.

6.3 Exclusive

Similar to the double module, the exclusive module consists of two optional detectors. An event is detected and saved in the log file only when the main detector is triggered and the excluding detector is not. If both detectors are triggered, an event is not created. A timing example is shown in Figure M.27.

7 Setting up the logger

To set up the logger, two aspects should be considered; timing settings and the type of output we get from RUBA. The common timing settings are related to

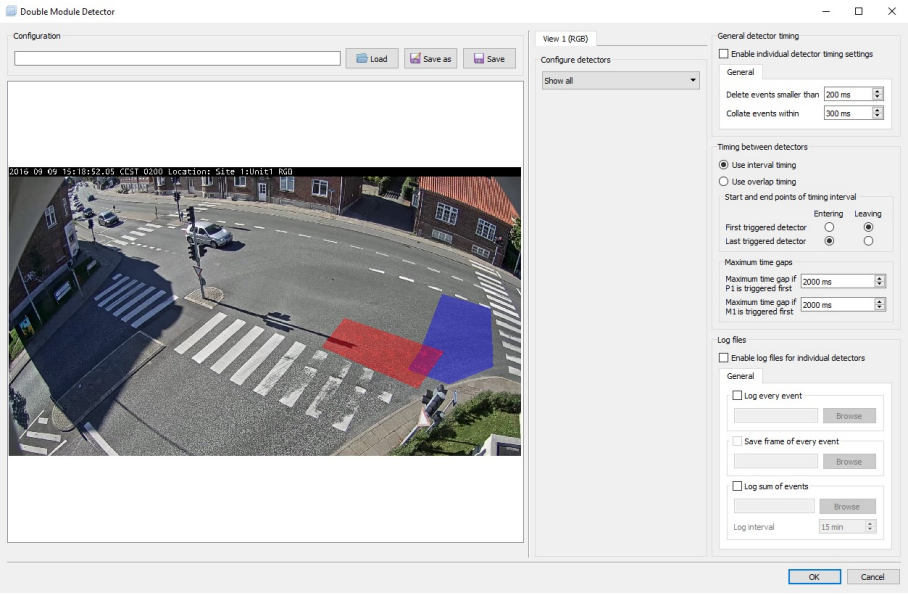


Fig. M.26: Example of a double module. The double module lets the user create two detectors of one's own choice per module.

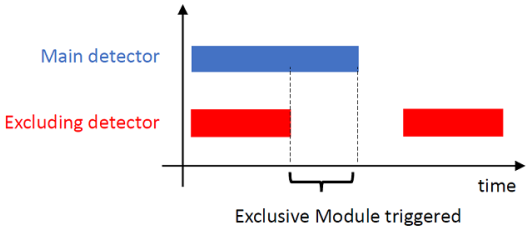


Fig. M.27: The Exclusive module is triggered when the main detector is triggered and the excluding detector is not.

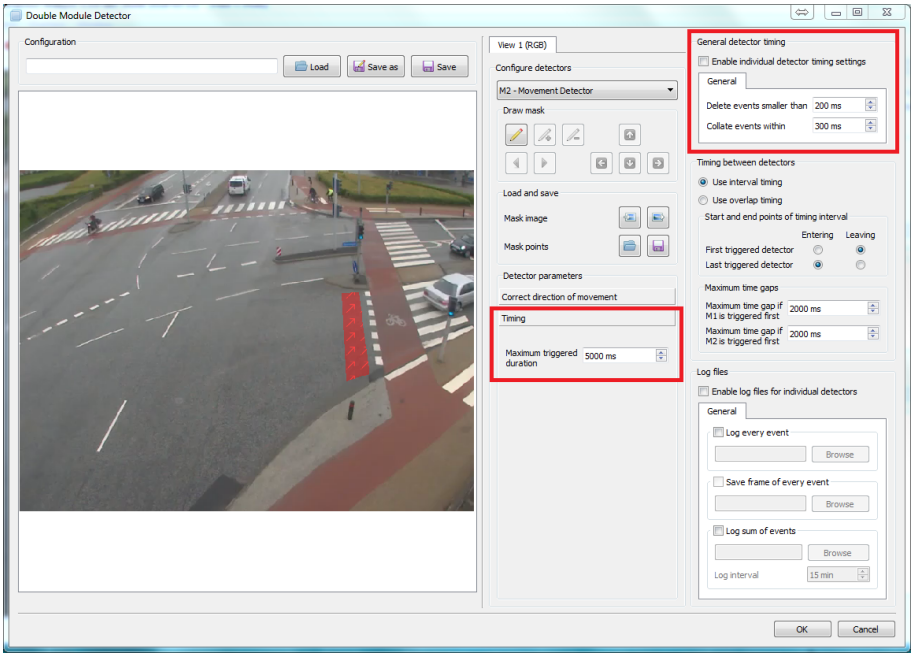


Fig. M.28: Common timing settings.

each individual detector. Furthermore, there are additional timing settings for double modules to define when an event should be detected.

7.1 Common timing settings

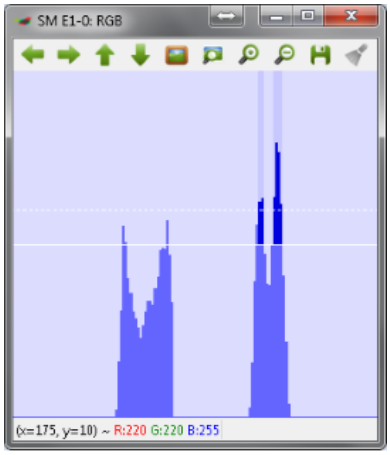
The common timing settings are used to adjust when an event should be written to the log. The fields for defining the common timing settings are marked in red on Figure M.28.

Delete events smaller than

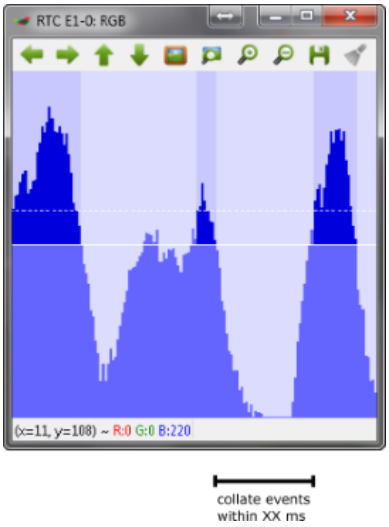
Deletes events that are only detected briefly, which often indicates noise in the image. Events with duration less than `Delete events smaller than` milliseconds will be omitted from the log. An example is shown in Figure M.29a.

Collate events within

Combines separate events into one event if the time gap between them is less than XX milliseconds. This protects against multiple detections of the same



(a) The two dark blue tops in the centre of the histogram will not be detected as an event if `Delete events smaller than` is greater than zero.



(b) The dark blue tops in the centre and the right of the image might be grouped as one event if the `Collate events within` setting is greater than the time difference between the two detections.

Fig. M.29: The common timing settings in RUBA.

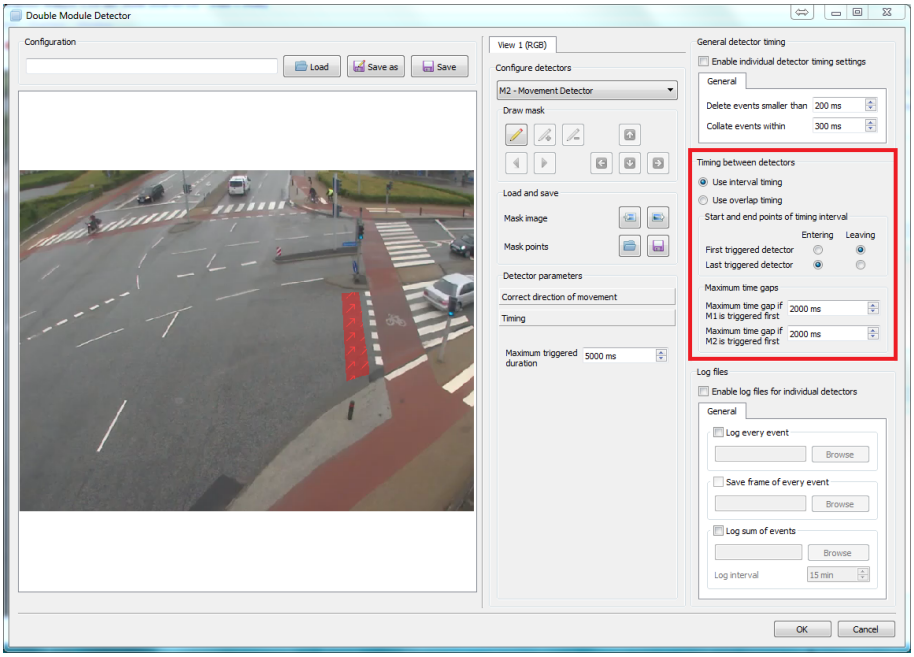


Fig. M.30: The Double Module offers two timing modes; interval timing and overlap timing.

object. If chosen too high, multiple road users driving close to each other will be registered as one road user. See Figure M.29b for an example.

Maximum triggered duration

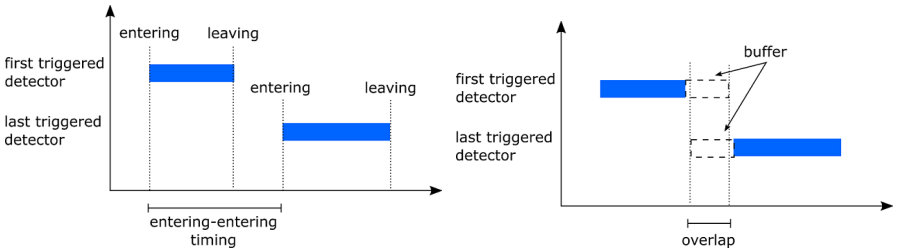
Defines the maximum allowed duration (in milliseconds) of an event. If an event is longer than the specified maximum duration, it will be cut off after the max triggered duration has gone, and a new event will be created immediately thereafter.

7.2 Double module timing settings

Timing between the two individual detectors of a double module can be defined using either *interval timing* or *overlap timing*, marked in red on Figure M.30.

Interval timing

The interval timing, illustrated in Figure M.31a, denotes the maximum accepted time gap (in milliseconds) from the detection of activity in one detector to the detection of activity in the other detector. The point for measuring the



(a) Illustration of interval timing. In this example, `Entering` is selected for both the `First triggered detector` and the `Last triggered detector`.

(b) Illustration of overlap timing.

Fig. M.31: Timing settings of the Double Module.

time gap can be either when the detector is activated (i.e. when the detector registers a that a road user enters the detector) or when the detector is left (i.e. when the road user has just left the detector).

Overlap timing

The overlap timing, illustrated in Figure M.31b, detects an event if both detectors are activated simultaneously. A buffer (in milliseconds) can be used to also log events where the two detectors are activated at almost the same time.

7.3 Logs

Three types of output can be created:

- **Log every event**: creates a .csv file with one line for each detection.
- **Save frame of every event**: saves an image of what triggered the detector. For double modules, the saved image contains a screenshot of what triggered each detector, put side by side.
- **Log sum of events**: creates a .csv file with the total number of detections per log interval(in minutes).

The log settings pane is shown in Figure M.32. The content of the log file depends on the detector module. Table M.5 gives an overview of the content of the log files.

Log Examples

Sample every event logs are shown in Figure M.33 and M.34. A sample of a sum of event log is shown in Figure M.35.

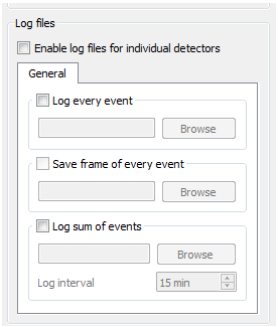


Fig. M.32: The log files settings pane.

	A	B	C	D	E	F
1	File	Date	Entering Detector 1	Leaving Detector 1	Duration	Frame
2	20170520_125729_6977.mkv	2017 05 20	2017 05 20 12:58:44.333	2017 05 20 12:58:44.666	333	M2-000001-2017-05-20-12-58-44.333-Entering-View-1.png
3	20170520_125729_6977.mkv	2017 05 20	2017 05 20 12:58:59.733	2017 05 20 12:59:00.100	367	M2-000002-2017-05-20-12-58-59.733-Entering-View-1.png
4	20170520_125729_6977.mkv	2017 05 20	2017 05 20 12:59:30.033	2017 05 20 12:59:30.433	400	M2-000003-2017-05-20-12-59-30.033-Entering-View-1.png
5	20170520_125729_6977.mkv	2017 05 20	2017 05 20 12:59:36.266	2017 05 20 12:59:36.666	400	M2-000004-2017-05-20-12-59-36.266-Entering-View-1.png
6	20170520_125729_6977.mkv	2017 05 20	2017 05 20 12:59:41.200	2017 05 20 12:59:41.633	433	M2-000005-2017-05-20-12-59-41.200-Entering-View-1.png
7	20170520_125729_6977.mkv	2017 05 20	2017 05 20 13:00:58.900	2017 05 20 13:00:59.233	333	M2-000006-2017-05-20-13-00-58.900-Entering-View-1.png
8	20170520_125729_6977.mkv	2017 05 20	2017 05 20 13:01:14.100	2017 05 20 13:01:14.433	333	M2-000007-2017-05-20-13-01-14.100-Entering-View-1.png
9	20170520_125729_6977.mkv	2017 05 20	2017 05 20 13:01:39.766	2017 05 20 13:01:40.100	334	M2-000008-2017-05-20-13-01-39.766-Entering-View-1.png
10	20170520_130230_03A2.mkv	2017 05 20	2017 05 20 13:02:37.833	2017 05 20 13:02:38.866	1033	M2-000009-2017-05-20-13-02-37.833-Entering-View-1.png
11	20170520_130230_03A2.mkv	2017 05 20	2017 05 20 13:03:16.900	2017 05 20 13:03:17.433	533	M2-000010-2017-05-20-13-03-16.900-Entering-View-1.png

Fig. M.33: Every event log from a Single Module Detector. The `Duration` column lists the time difference in milliseconds between the `Entering Detector 1` and `Leaving Detector 1`.

	A	B	C	D	E	F	G	H	I	J	K
1	File	Date	Entering Detector 1	Leaving Detector 1	Duration	Entering Detector 2	Leaving Detector 2	Duration	Time Gap 1-2	First Triggered	Frame
2	20170521_073023_B63B.mkv	2017 05 21	2017 05 21 07:34:53.633	2017 05 21 07:34:53.833	200	2017 05 21 07:34:53.966	2017 05 21 07:34:54.866	900	333	M1	DM-000001-2017-05-21-07-34-53.633-Entering-Entering-View-1.png
3	20170521_085036_140C.mkv	2017 05 21	2017 05 21 08:52:25.133	2017 05 21 08:52:25.633	500	2017 05 21 08:52:25.366	2017 05 21 08:52:26.000	634	233	M1	DM-000002-2017-05-21-08-52-25.133-Entering-Entering-View-1.png
4	20170521_095045_23BC.mkv	2017 05 21	2017 05 21 09:52:28.466	2017 05 21 09:52:29.100	634	2017 05 21 09:52:28.633	2017 05 21 09:52:29.133	500	167	M1	DM-000003-2017-05-21-09-52-28.466-Entering-Entering-View-1.png
5	20170521_133622_700D.mkv	2017 05 21	2017 05 21 13:38:43.333	2017 05 21 13:38:49.400	6067	2017 05 21 13:38:46.300	2017 05 21 13:38:46.800	500	2967	M1	DM-000004-2017-05-21-13-38-43.333-Entering-Entering-View-1.png
6	20170521_151638_DDDA.mkv	2017 05 21	2017 05 21 15:20:09.733	2017 05 21 15:20:11.000	1267	2017 05 21 15:20:14.000	2017 05 21 15:20:14.866	866	4267	M1	DM-000005-2017-05-21-15-20-09.733-Entering-Entering-View-1.png
7	20170522_075025_E1C4.mkv	2017 05 22	2017 05 22 07:54:40.633	2017 05 22 07:54:42.333	1700	2017 05 22 07:54:44.566	2017 05 22 07:54:45.200	634	3933	M1	DM-000006-2017-05-22-07-54-40.633-Entering-Entering-View-1.png
8	20170522_172658_B648.mkv	2017 05 22	2017 05 22 17:30:44.766	2017 05 22 17:30:45.733	967	2017 05 22 17:30:49.433	2017 05 22 17:30:50.100	667	4667	M1	DM-000007-2017-05-22-17-30-44.766-Entering-Entering-View-1.png
9	20170522_184710_D513.mkv	2017 05 22	2017 05 22 18:51:23.333	2017 05 22 18:51:27.166	3833	2017 05 22 18:51:26.766	2017 05 22 18:51:27.566	800	3433	M1	DM-000008-2017-05-22-18-51-23.333-Entering-Entering-View-1.png
10	20170523_070518_2767.mkv	2017 05 23	2017 05 23 07:08:14.900	2017 05 23 07:08:16.033	1133	2017 05 23 07:08:19.233	2017 05 23 07:08:19.733	500	4333	M1	DM-000009-2017-05-23-07-08-14.900-Entering-Entering-View-1.png
11	20170523_072521_44E6.mkv	2017 05 23	2017 05 23 07:27:11.300	2017 05 23 07:27:13.233	1933	2017 05 23 07:27:15.400	2017 05 23 07:27:16.833	1433	4100	M1	DM-000010-2017-05-23-07-27-11.300-Entering-Entering-View-1.png
12	20170523_082530_23B7.mkv	2017 05 23	2017 05 23 08:29:20.833	2017 05 23 08:29:21.766	933	2017 05 23 08:29:21.233	2017 05 23 08:29:21.633	400	400	M1	DM-000011-2017-05-23-08-29-20.833-Entering-Entering-View-1.png

Fig. M.34: Every event log from a Double Module Detector. In the `First Triggered` column, we see that the Movement Detector 1 (M1) always is triggered first in this sample. The `Frame` column shows the file name of the corresponding snapshot image for each event. If this column is empty, the `Save frame of every event` checkbox has not been ticket in the log settings.

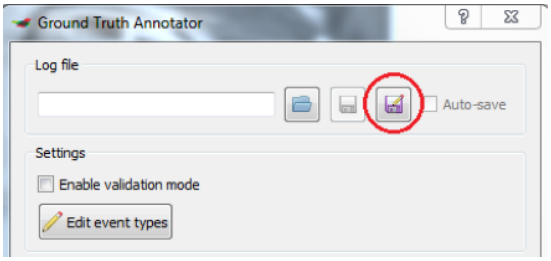
	A	B	C	D	E
1	File	Date	TimeStart	TimeEnd	Detections
2	20170520_125729_6977.mkv	2017 05 20	2017 05 20 12:57:29.000	2017 05 20 13:02:29.000	8
3	20170520_125729_6977.mkv	2017 05 20	2017 05 20 13:02:29.000	2017 05 20 13:07:29.000	6
4	20170520_130230_03A2.mkv	2017 05 20	2017 05 20 13:07:29.000	2017 05 20 13:12:29.000	13
5	20170520_130731_C632.mkv	2017 05 20	2017 05 20 13:12:29.000	2017 05 20 13:17:29.000	13
6	20170520_131232_A3A0.mkv	2017 05 20	2017 05 20 13:17:29.000	2017 05 20 13:22:29.000	9
7	20170520_131732_44BC.mkv	2017 05 20	2017 05 20 13:22:29.000	2017 05 20 13:27:29.000	4
8	20170520_132233_D114.mkv	2017 05 20	2017 05 20 13:27:29.000	2017 05 20 13:32:29.000	7
9	20170520_132734_AE80.mkv	2017 05 20	2017 05 20 13:32:29.000	2017 05 20 13:37:29.000	6
10	20170520_133235_9998.mkv	2017 05 20	2017 05 20 13:37:29.000	2017 05 20 13:42:29.000	7
11	20170520_133736_4BC2.mkv	2017 05 20	2017 05 20 13:42:29.000	2017 05 20 13:47:29.000	5

Fig. M.35: Sum of event log. All detector modules produce sum logs in this format.

Table M.5: Contents in the log files. The files `Analytics` are created when marking `Log every event` in the settings/drawing window of the detector, while `Counts` will be created when marking `Log sum of events`. In the latter, the desired time interval can be specified (in minutes), e.g. 15 minutes. *Legend*: Single Module = SM, Double Module = DM, Exclusive Module = XM.

	Log every event		Log sum of events	Description
	SM, XM	DM	All modules	
File	x	x	x	File name of the video
Date	x	x	x	Date for video recording
Entering	x	x		Time stamp for arrival to the detector (i.e. there is activity in the detector)
Leaving	x	x		Time stamp for when the detector has been left (i.e. is empty again)
Timegap 1-2		x		Time difference between an object has triggered detector 1 and an object has triggered detector 2
Timegap DetectorX		x		Time from one object arrives to detector X (trigger the detector) to the road user has left the detector
FirstTriggered		x		Which of the two detectors was triggered first?
TimeStart			x	Time for the beginning of the time interval
TimeEnd			x	Time for the end of the time interval
Object			x	Number of objects that have been detected within a specific time interval

(a) Opening the Ground Truth Annotator



(b) Save log files automatically in Ground Truth Annotator by pressing the `Auto-save` button.

Fig. M.36: Ground Truth Annotator

8 Ground Truth Annotator

RUBA features a built-in option to perform manual event-based annotation based on timestamps. For example, we might want to annotate whenever a car turns right in an intersection or whenever a pedestrian passes a specific line in a zebra crossing. In the example below, we will set up the Ground Truth Annotator to perform annotation of different road users at an intersection.

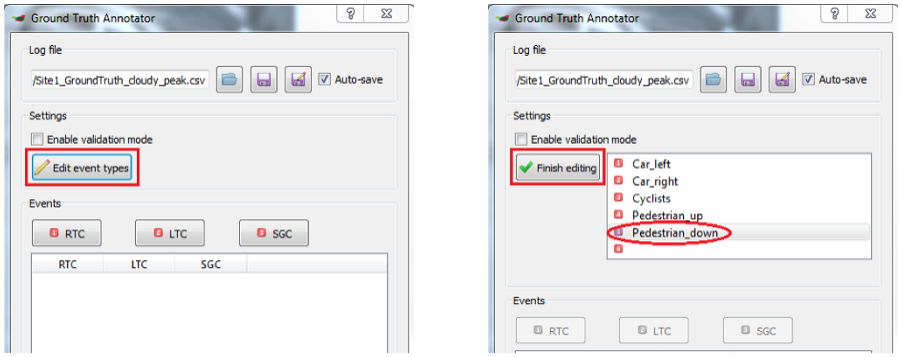
Click on Ground Truth Annotator in the main RUBA menu, marked with red on Figure M.36a.

A new window opens. Click on `Save log file as`, shown in Figure M.36b, and specify the name of the file and where to save the log file. Put a check mark next to Auto-save to save the log automatically.

Click on Edit event types to specify the types of road users to register. Double click on the name to change, and press `Finish editing` to include the event types in the `Event` panel. This process is also illustrated in Figure M.37.

Adjust the window and column sizes to get it to look more clear and place the window as shown in Figure M.38 so that you can see the window and the main window of RUBA at the same time.

Click in the Ground Truth Annotator (GTA) window and press space to start. Press space again to pause playback. Click on the corresponding button in the GTA window or press the number on the keyboard every time a road user of that particular type passes. Make sure that you register all road users at the same spot every time. All road users must be counted individually in the tool, so if there is a group of 2 pedestrians, press twice to register the road users.



(a) **(b)**

Fig. M.37: Editing the event types of the Ground Truth Annotator

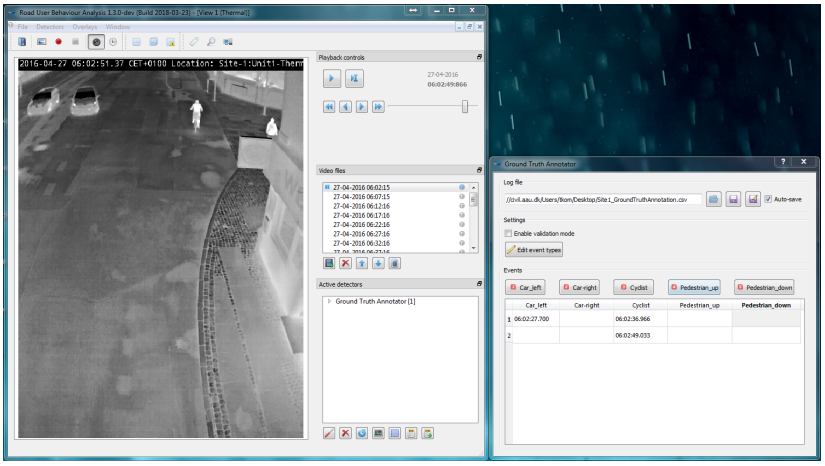


Fig. M.38: The Ground Truth Annotator should be placed beside the video window in RUBA to make the annotation process feasible.

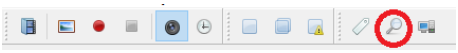


Fig. M.39: Opening the Log File Reviewer.

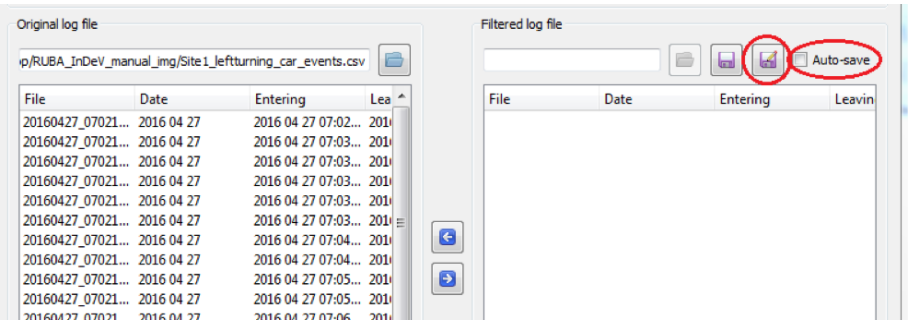


Fig. M.40: Setting the file path of the filtered log file.

8.1 Reviewing annotations

You may continue the annotation process or review the previous annotations by loading them into the Ground Truth Annotator. Either start the video from the beginning or double-click on an annotated time stamp to jump to this particular point in the video.

If you tick the `Enable validation mode` button during playback, the most recent annotated time stamp will be selected in the `Events` pane. If the `Show masks` button is also enabled in the main RUBA window, the annotated event information will be overlayed on the video when it takes place.

9 Log File Reviewer

The Log File Reviewer is a tool to review and validate the log files as output by the detector modules of RUBA. Open the Log File Reviewer by pressing the button on the toolbar, illustrated in Figure M.39, or press `F11`.

Open the log file that you want to inspect by clicking the folder icon in the `Original log file` window, shown on the left of Figure M.40. Hereafter, click on the save button in the `Filtered log file window` and specify where to save the (validated) log file for further processing. Tick off the check box `Auto-save`.

Define how many seconds should be played before and after the time stamp, e.g. 15 sec before the `Entering` and 10 sec after the `Leaving` timestamps in the log files. An example is shown in Figure M.41.

Adjust the size of the RUBA windows as illustrated in Figure M.42 so that the normal window and the log file reviewer are both visible and placed next

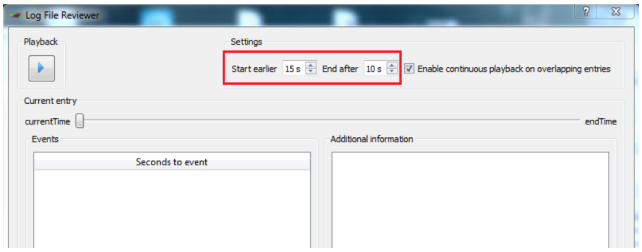


Fig. M.41: Setting the video playback properties.

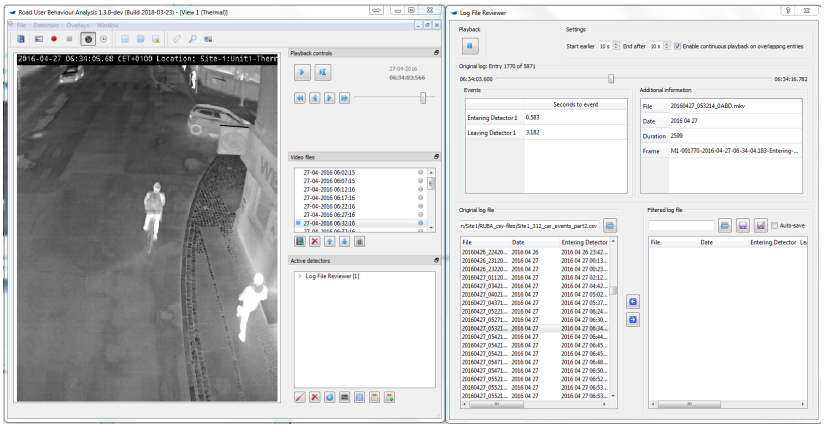


Fig. M.42: The Log File Reviewer next to the main RUBA window.

to each other.

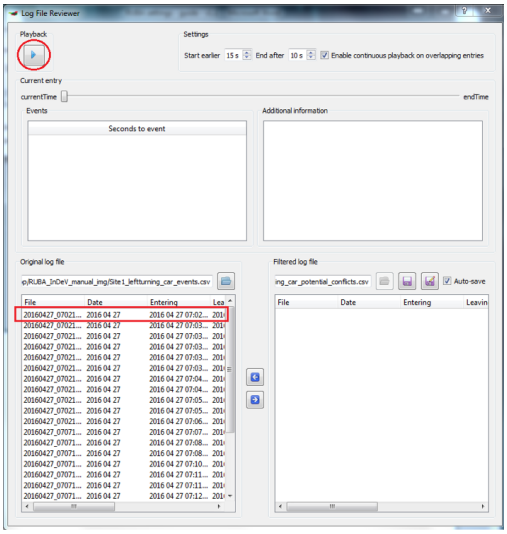
Double click on the first video in the list, then click the playback button, marked in red on Figure M.43a. All events will be played one by one without break. Press the pause button to pause the playback (keyboard shortcut: space) and press again to start the playback (keyboard shortcut: space). You can double click on any of the events in the list to go to that event.

Adjust the speed on the sliding bar (Figure M.43b) if it goes too fast/slow. Do the following to insert and delete events in the filtered log file:

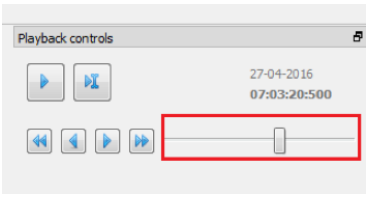
1. Click on the blue arrow pointing right (keyboard shortcut: INS) to put an event from the original when there is a potential conflict. It is possible to select more events at once.

2. Click on the blue arrow pointing left (keyboard shortcut: DEL) to remove an event from the filtered log file. These arrows are marked in red on Figure M.44.

Please note that only the filtered log file is altered during this process. The original log file is read-only.



(a) Starting the playback based on the events of the log file.



(b) Adjusting the playback speed in the main RUBA window.

Fig. M.43: Playback in the Log File Reviewer

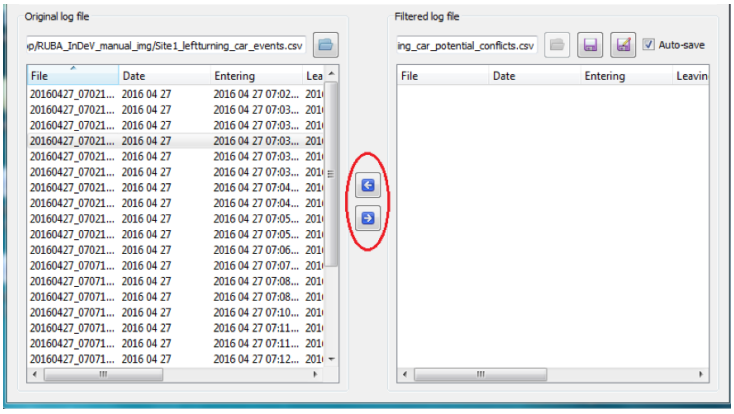


Fig. M.44: Inserting and deleting events into the filtered log file.

Part IV

Dissemination Activities

Paper N

Deep Learning - et gennembrud indenfor kunstig intelligens

Morten B. Jensen, Chris H. Bahnsen, Kamal Nasrollahi and Thomas B. Moeslund

The paper has been published in
Aktuel Naturvidenskab Vol. 2, pp. 8–13, 2018.

The layout has been revised.

I løbet af de seneste 10 år er kunstige, neurale netværk gået fra at være en støvet, udstødt teknologi til at spille en hovedrolle i udviklingen af kunstig intelligens. Dette fænomen kaldes deep learning og er inspireret af hjernens opbygning.

1 Introduction

Hvordan kan en computer vinde over verdensme- steren i GO, hvor der er flere mulige kombinationer på spillepladen end atomer i universet? Hvordan kan en bil forstå, at der er en fodgænger foran den og selv bremse?

Svaret på denne type spørgsmål er intelligente computersystemer, der lærer ved at analysere data – rigtig meget data. Den nyeste metode indenfor dette forskningsområde kaldes deep learning. Metoden har på få år revolutioneret store dele af forskningsverdenen og er nu på vej ud i alle grene af samfundet, hvor den forventes at få afgørende betydning.

Et gammelt ordsprog siger, at viden er magt! Måske er data et af de vigtigste elementer i dannelsen af viden, men hvordan man styrer og udnytter data, er endnu vigtigere. Derfor har forskere altid forsøgt at udvikle avancerede måder til indsamling af data for derefter at udnytte det bedst muligt. For at finde inspiration til at udvikle bedre databehandlingsteknikker har forskere kigget på hjernens opbygning og opførsel i håb om at kunne opnå en forståelse, der efterfølgende kan implementeres i computere. Dette forskningsområde

kendes også som kunstig intelligens. På grund af hjernens komplekse struktur har det altid været meget udfordrende at forstå hjernens grundlæggende funktionalitet for derefter at opbygge et hjerne-lignende system. På trods af, at ingeniører længe har formået at konstruere systemer, der kan ef- terligne hjernen ved simple opgaver, så har forskere stødt hovedet mod muren, når det kom til at konstruere systemer, der er i stand til at løse mere udfordrende opgaver, for eksempel genkendelse af objekter.

Imidlertid har nylige fremskridt inden for dataindsamling og rå processeringskraft gjort det muligt at bygge systemer baseret på kunstig intelli- gens, der kan løse komplekse problemer som objektdetektion, genkendelse og tracking. Systemerne er nu så gode, at de i nogle tilfælde klarer sig bedre end menneskelige eksperter.

Disse systemer bliver trænet ved hjælp af massive datamængder gennem matematiske algoritmer, der er bedre kendt under paraplybetegnelsen deep learning. Før vi kommer nærmere ind på det, må vi en tur omkring hjernen for at få en grundlæggende forståelse af disse systemer.

2 Hjernen

Hjernen er en af de mest komplekse strukturer, vi kender. Den er opbygget af 100 milliarder celler kaldet neuroner, og der er cirka samme antal neuroner i hjernen, som der er stjerner i Mælkevejen. I figur N.1 ses en illustration af et neuron. Hvert neuron har: Et cellelegeme, indeholdende kernen, som er neuronets behandlingscenter. Et sæt indgangsforbindelser, dendritter, som bringer signaler fra de andre neuroner til kernen i det nuværende neuron. En axon, som overfører resultaterne af behandlingen af indgangssignalerne i kernen til de neuroner, der er forbundet til det aktuelle neuron via sine udgangsforbindelser (axonterminaler).

En gruppe af disse små hjerneneuroner, der er internt forbundet med hinanden, er ansvarlige for at udføre en specifik opgave. For eksempel udføres matematiske operationer i en bestemt del af hjernen, mens følelser opfattes af en anden gruppe neuroner. Ved løsning af specifikke opgaver viser de ansvarlige grupper af neuroner mere elektrisk aktivitet end resten af hjernen. Disse elektriske aktiviteter skyldes frigivelse af kemiske stoffer mellem neuronerne, der er internt forbundet med hinanden. Hvis summen af kemiske stoffer ved neuronet er større end et bestemt niveau, bliver neuronet aktiveret. I modsat fald forbliver det passivt.

Når vi som menneske prøver at lære en bestemt opgave, for eksempel når en baby lærer at gå, gennemføres denne læring gennem adskillige forsøg. Under disse forsøg lærer hjernen, eller rettere: En specifik gruppe neuroner lærer, hvordan de skal aktiveres for at udføre den specifikke opgave. Mængden

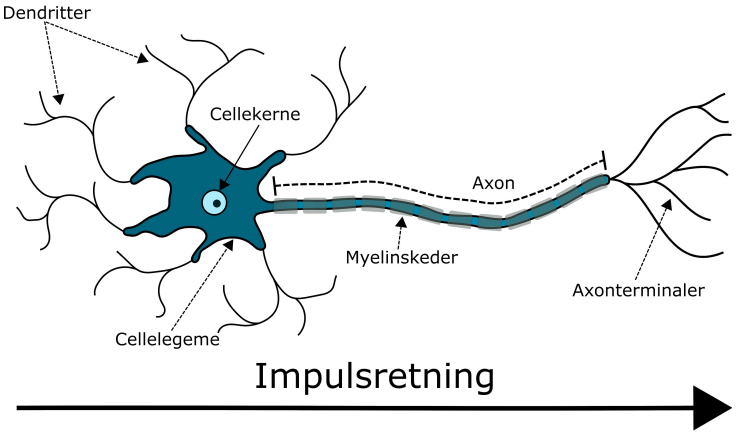


Fig. N.1: Illustration af et neuron, som er hjernens byggesten.

af de kemiske stoffer, der frigives mellem neuronerne, definerer graden af forbindelse, også kaldet vægtningen, mellem de tilsluttede neuroner.

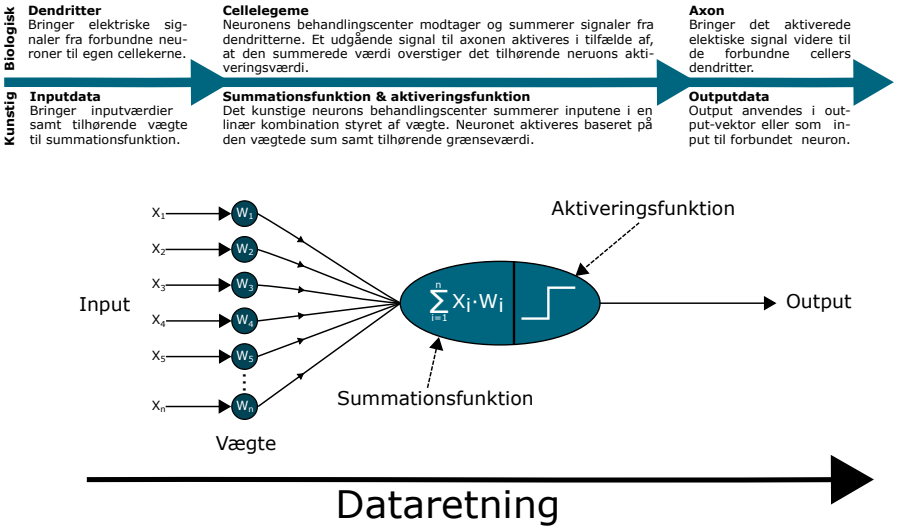


Fig. N.2: Princippet i en kunstig neuron.

Man kan simulere det biologiske neuron med en matematisk funktion, der består af en lineær kombination af alle inputs til neuronet. Den lineære kombination styres af vægtene af de enkelte inputs. Denne sum af vægtede input

svarer til mængden af de kemiske stoffer, der kommer til et neuron. Derefter bestemmer en såkaldt aktiveringsfunktion, om neuronet skal aktiveres eller forblive passivt. Hvis den vægtede sum er større end en given grænseværdi, aktiveres neuronet. Dette princip er illustreret i figur N.2.

Et kunstigt neuralt netværk består af en kombination af disse kunstige neuroner i forskellige lag, der er internt forbundet med hinanden gennem vægtede forbindelser. Antallet af lag beskriver dybden af netværket. Man betegner et neuralt netværk som dybt, hvis det indeholdertre eller flere lag.

Aktiveringsfunktioner spiller en nøglerolle i kunstige neurale netværk. Hvis aktiveringsfunktionen udelukkende består af lineære funktioner, kan det kunstige neurale netværk udelukkende beskrive lineære fænomener, og dets samlede funktion kan grundlæggende beskrives af én stor matrix. Hvis aktiveringsfunktionen derimod ikke kan beskrives som en lineær kombination af dens input, er det kun dybden af det kunstige neurale netværk, der begrænser kompleksiteten af de funktioner, som netværket kan beskrive.

3 Læring

For at lære et kunstigt neuralt netværk at udføre en specifik opgave kræves en læringsalgoritme, hvis formål er at finde de rette vægte mellem netværkets neuroner. Vægtene læres gennem adskillige iterationer, hvor det neurale netværk præsenteres for store mængder træningsdata. Hver enkelt stykke data er annoteret, det vil sige, at det er parret med den ønskede respons fra det neurale netværk – for eksempel at et billede af en hund hører til kategorien "hund", hvis formålet med det neurale netværk er at genkende objekter i billeder. Når billedet er kørt igennem det neurale netværk, giver netværket dets bud på hvilken kategori, billedet tilhører. Herefter udregnes forskellen mellem det beregnede og det ønskede resultat, hvilket kaldes krydsentropitabet. Det beregnede krydsentropitab fødes herefter baglæns ind i det neurale netværk og opdaterer vægtene i retning af det ønskede resultat.

I starten resulterer det neurale netværk ikke i andet end støj. Men ganske langsomt, iteration for iteration, lærer netværket at tilpasse sig det pågældende træningsdata. Når det beregnede resultat konvergerer mod det ønskede resultat, er træningen afsluttet og netværket er nu specialiseret i at klassificere datasættet. Hvis datasættet indeholder tilstrækkeligt mange annoterede billeder og er repræsentativt for de ønskede kategorier, for eksempel hunde og katte, har man nu en udmærket hunde- og kattedetektor.

Et netværk trænes ved at udregne dets respons for en række billeder (grønne pile i figur N.3), hvor vi på forhånd har defineret det ønskede resultat (annoteret data). Forskellen mellem det ønskede resultat og det beregnede resultat beregnes i det såkaldte krydsentropitab, som føres baglæns gennem

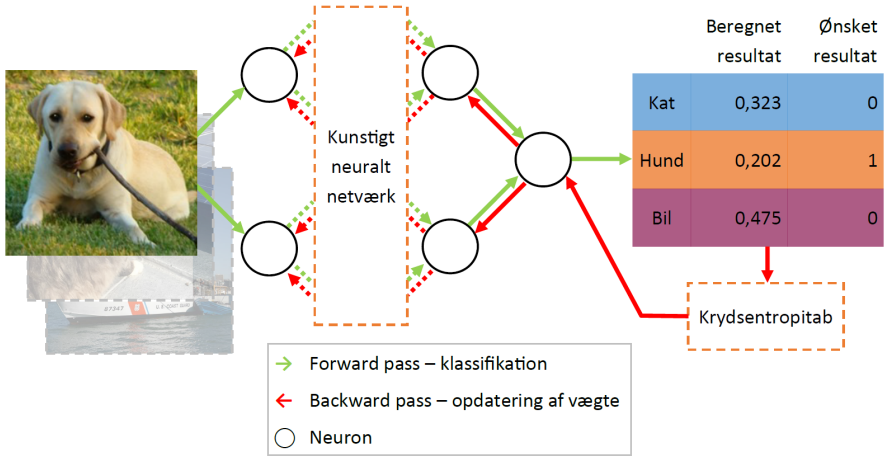


Fig. N.3: Træningsproces af et kunstigt neuralt netværk.

netværket for at opdatere dets vægte (røde pile i figur N.3).

To af nøgleordene bag denne læringsalgoritme er differentiabilitet og kædereglen for diffentiering af sammensatte funktioner. Alle de neuroner, som et kunstigt neuralt netværk er sammensat af, er grundlæggende (stykvist) differentierbare funktioner. Det betyder, at vi kan flytte det samlede netværks opførsel ved, neuron for neuron, at finde gradienten for den partielt differentierede funktion, opdatere funktionens vægte på baggrund heraf, og føre gradienten videre til de neuroner, som funktionen er forbundet til. Denne proces gentages for hver iteration, indtil alle neuroner er opdateret.

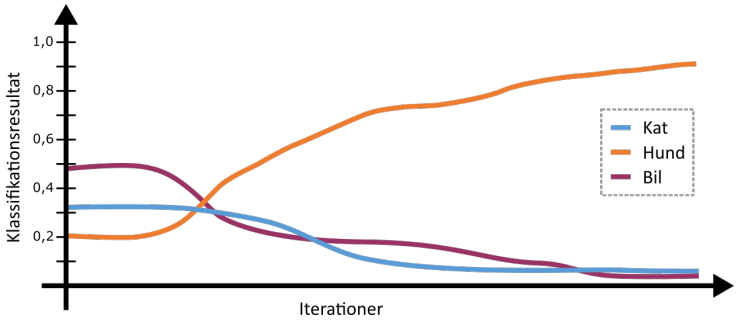


Fig. N.4: Iterativ læring af et kunstigt neuralt netværk. I starten resulterer netværket ikke i andet end støj, men jo flere annoterede billeder, der køres igennem netværket, jo bedre bliver det til at klassificere billeder af hunde som "hund".

4 Hvorfor først nu?

Kunstige neurale netværk går tilbage til 1940'erne. Disse netværk var imidlertid ikke særlig populære i samtiden på grund af den beregningsmæssige kompleksitet og manglende træningsdata.

Den beregningsmæssige kompleksitet skyldes, at et netværk, der kan løse praktiske problemer, indeholder mindst tre lag med mange neuroner i hvert lag, der er internt forbundet med hinanden. Denne type netværk er kendt som en Multi-Layer Perceptron (MLP). I en MLP er hvert neuron i et lag forbundet til alle neuroner i det næste lag af netværket, som set i figur N.5. Dette fænomen, der er kendt som fuldt forbundne netværk, resulterer i store matricer, der beskriver vægtningen af neuronernes forbindelser. De store matricer fører igen til beregningsmæssigt krævende læringsalgoritmer, der i mange år var for store til, at computere kunne håndtere dem. Dette ændrede sig dog med introduktionen af såkaldte Graphics Processing Units (GPU) i 1990'erne, som tilbød hurtig og parallel databehandling. Brugen af GPU'er har gjort det muligt at implementere neurale netværk i praksis, hvorefter deres popularitet kun er steget. Faktisk er neurale netværk nu blandt de allerbedste værktøjer, der er i stand til at løse meget komplicerede problemer som billedbaseret objektgenkendelse. Denne succes skyldes imidlertid ikke kun udviklingen af bedre GPU'er, men også tilgængeligheden af enorme mængder data.

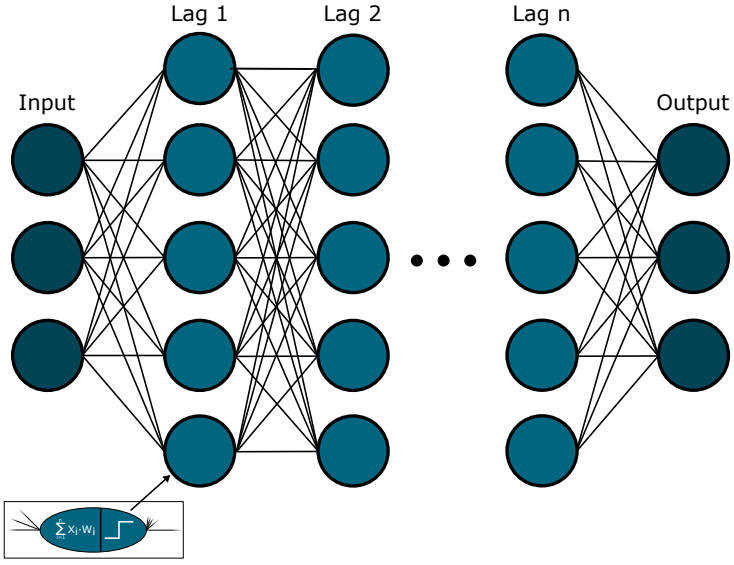


Fig. N.5: I et Multi-Layer Perceptron (MLP) netværk er alle neuroner forbundet til hinanden imellem lagene.

Da kunstige neurale netværk udelukkende kan lære på baggrund af eksempler, er tilgængeligheden af eksempeldata kritisk. Jo mere data, jo bedre er læringsprocessen. Imidlertid var store databaser ikke så almindelige for blot 10 år siden. Men siden 2010 er enorme databaser gradvist blevet opbygget. Et eksempel er ImageNet, der består af cirka 14 millioner annoterede billeder, inddelt i mere end 20.000 forskellige kategorier.

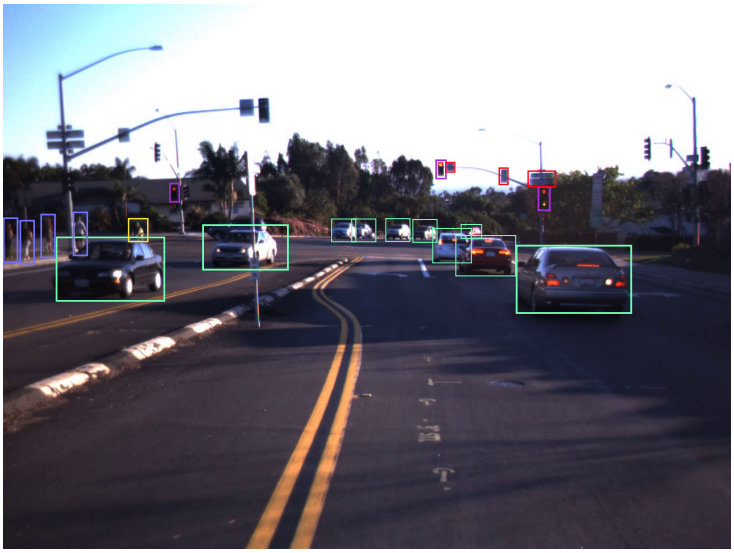


Fig. N.6: Overlejrede annoteringer af objekter i trafikken, hvor hver farve indikerer en kategori. Denne type data er fx vigtig for træningen af selvkørende biler.

De fleste billeder i ImageNet indeholder kun én annotering, det vil sige at hele billedet tilhører én kategori. En anden og mere omfattende måde at annotere billederne på er at definere det specifikke område i billedet, der indeholder et givent objekt – for eksempel fodgængere, cyklister, biler og trafikskilte, som ses i figur N.6.

5 Fra machine learning til deep learning

Traditionelle machine learning-teknikker er baseret på at udvikle og udvælge specifikke karaktertræk, også kaldet features, ved de objekter, man ønsker at finde og genkende. Det har betydet, at forskere tidligere har brugt tid på manuelt at definere features, som efter deres vurdering var unikke og gav en god repræsentation af de ønskede objekter.

Til detektion af trafikskilte vil man i traditionel machine learning udvælge features, der kan beskrive skiltets cirkulære form og dets karakteristiske røde kant. Herefter udvælger man manuelt en eller flere metoder, der kan

konvertere de ønskede features til en matematisk repræsentation. Disse features bruges til at træne en machine learning-algoritme, som benytter de udregnede features til at skelne mellem trafikskilte og ikke-trafikskilte.

Til trods for, at det i sidste ende er computerens algoritmer, der udregner det endelige resultat, indebærer traditionel machine learning en del manuelt arbejde med at definere hvilke features, der er relevante. Deep learning har til forskel fra machine learning ingen behov for menneskelig indblanding i forbindelse med udvælgelse og udformning af features. Sammenhængen mellem kunstig intelligens, machine learning og deep learning ses illustreret i figur N.7.

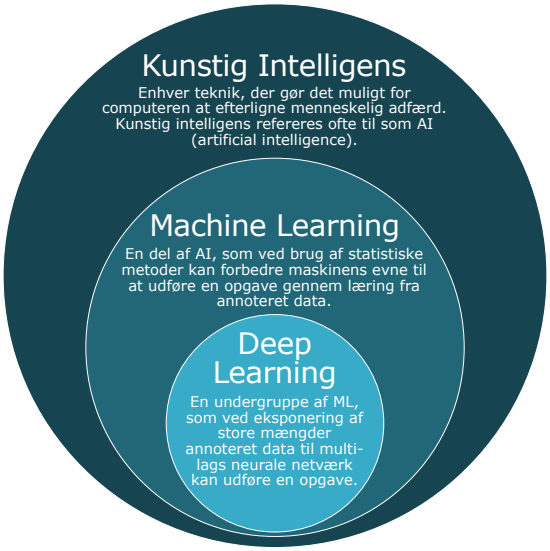


Fig. N.7: Sammenhæng mellem kunstig intelligens, machine learning og deep learning.

Det kræver dog stadig menneskelig indblanding, når deep learning-netværkene skal designes. Det gøres for eksempel ved at definere netværkets størrelse, det vil sige hvor mange lag, netværket skal bestå af. Workflowet fra mellem machine learning og deep learning er illustreret i figur N.8.

Et lag består af en række funktioner. Den vigtigste funktion i moderne neurale netværk er en såkaldt convolution (på dansk en foldning), og derfor kaldes disse netværk også Convolutional Neural Networks (CNN'er). Convolution er en matematisk operation, der benytter sig af et filter. Filtrenes overordnede funktion er at trække features ud af inputbilledet, og et moderne neuralt netværk indeholder rigtig mange filtre, der er grupperet i flere convolution-lag. Populære deep learning-netværk som AlexNet, VGG, GoogLeNet og Microsoft ResNet er alle CNN'er, og de indeholder henholdsvis 8, 19, 22 og 152 lag. Et eksempel på en convolution ses til venstre i figur N.9,

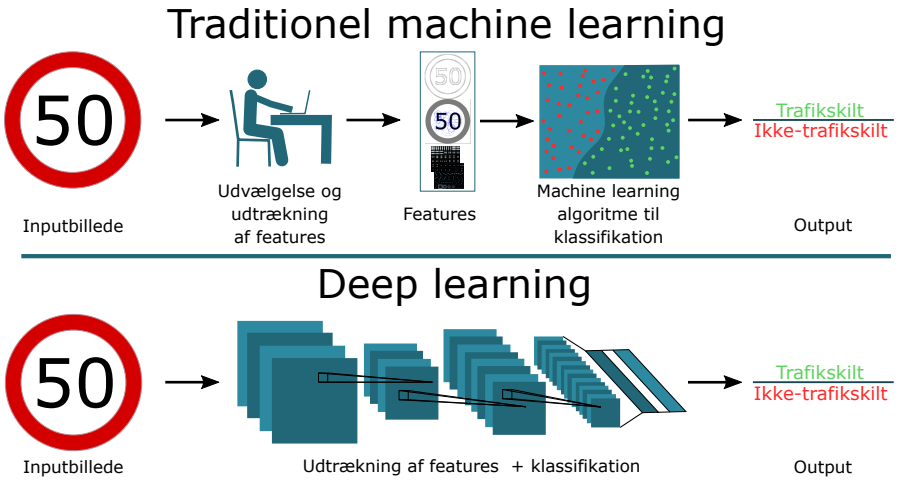


Fig. N.8: Sammenligning af workflowet for traditionel machine learning og deep learning. I modsætning til traditionel machine learning kræver deep learning ikke manuel udvælgelse af features.

hvor der benyttes et filter af størrelsen 3x3 pixels med udgangspunkt i den pixel, der er markeret med rød ring. Convolution består i, at man anvender 3x3 filteret på den "røde" centerpixel samt dets nabopixels, illustreret med det grå område i input- matricen. Den resulterende pixelværdi i outputmatricen opnås ved at gange filterets vægte på de tilhørende pladser i inputmatricen for derefter at summere resultatet. Herefter rykker vi vores 3x3 filter én gang til højre og gentager udregningen.

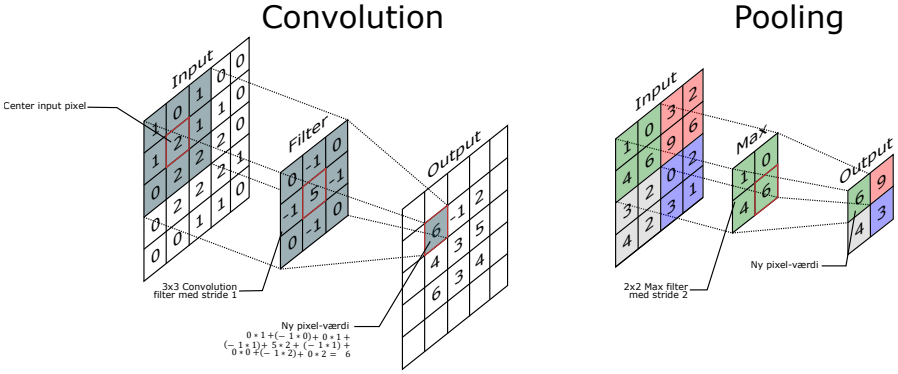


Fig. N.9: De vigtigste funktioner i moderne neurale netværk er convolution og pooling.

Outputtet fra convolution kaldes et feature map, og det er udgangspunktet for et andet meget anvendt lag i neurale netværk kaldet pooling, som ses til højre i figur N.9. I eksemplet på pooling bruges her et såkaldt max pooling-

filter med en størrelse på 2x2 pixels og en stride på 2 pixels. En stride på 2 pixels betyder, at vi flytter filteret 2 pixels til højre efter hver operation. Maxfilteret undersøger alle værdierne i et 2x2 område, tager den højeste værdi heri og smider de øvrige værdier væk. Den højeste værdi udgør nu den nye pixelværdi i outputmatricen. Denne funktion reducerer størrelsen på de feature maps, der genereres fra convolution-laget, således at man opnår en mere kompakt repræsentation.

En af årsagerne til, at CNN fungerer så godt, er, at netværkene selv kan lære at sammensætte både simple og komplekse features i deres convolution-lag – uden at man som operatør specifikt beder dem om at gøre sådan. Eksempel på både simple såvel som komplekse features ses i figur N.10.

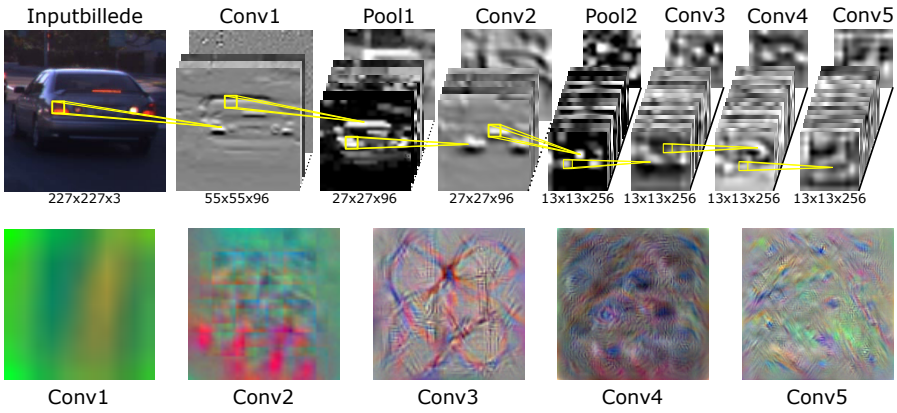


Fig. N.10: Eksempel på et AlexNet-inspireret neuralt netværk med 5 convolution-lag og 2 pooling-lag samt dertil hørende lærte features for hvert convolution-lag (øverst). Conv-lagene ændrer sig gradvis fra at være ret genkendelige omrids af bilen i conv1 til mere komplekse strukturer i conv5, som knap nok er genkendelige for det menneskelige øje. Den nederste del af figuren visualiserer, hvad 1 tilfældigt udvalgt feature-map i hvert af de 5 convolution-lag reagerer på i billedet. Conv1-features repræsenterer kanter og farveforskelle i forskellige retninger, mens de senere convolution-lag repræsenterer komplekse, specialiserede mønstre.

6 Ikke begrænset af menneskelige sanser

Det ser i store træk ud til, at deep learning og kunstige neurale netværk virker som hjernen ved løsning af bestemte opgaver. Det betyder, at kunstig intelligens i princippet vil kunne klare det samme som et menneske. Potentialet er dog endnu større for den kunstige intelligens, da dets input ikke er begrænset til de menneskelige sanser, men vil kunne opfatte verden gennem et utal af sensorer og have adgang til ufattelige mængder information. For at nå så langt kræves der dog betydelige fremskridt indenfor udvikling af læringsalgoritmer og håndtering af information. På vejen dertil vil deep learning ændre fremti-

den i mange forskellige applikationer og sektorer, fra sikkerhed og finans til medicin og transport. Vi er glade for at være en del af dette spændende eventyr.

ISSN (online): 2446-1628
ISBN (online): 978-87-7210-358-7

AALBORG UNIVERSITY PRESS