Automatic Video-based Analysis of Human Motion

Fihl, Preben

Publication date:
2011

Document Version
Accepted author manuscript, peer reviewed version

Media Technology

Ph.D. Dissertation

Automatic Video-based Analysis of Human Motion

Preben Fihl

Faculty of Engineering and Science

Aalborg University 2011

About the Author

Preben Fihl received the M.Sc.EE degree in 2005 and the Ph.D. degree in 2011, both from Aalborg University, Denmark. He is working primarily with computer vision with a strong focus on video-based analysis of humans and their motion. He is currently working as an assistant professor at the department of Architecture, Design, and Media Technology, Aalborg University.

Automatic Video-based Analysis of Human Motion

A Ph.D. dissertation
by
Preben Fihl

Department of Arcihtecture, Design, and Media Technology
Faculty of Engineering and Science
Aalborg University, Denmark
E-mail: fihl@create.aau.dk
URL: http://www.cvmt.dk/∼pfa

January 2011

ISBN 978-87-992732-4-9 (printed version)
Electronic version: `http://bit.ly/vbn-aau-dk-fihl-phd_thesis`

This dissertation was submitted in January 2011 to the Faculty of Engineering and Science, Aalborg University, Denmark, in partial fulfilment of the requirements for the Doctor of Philosophy degree.

The defence took place at Aalborg University, Niels Jernes Vej 14, DK-9220 Aalborg on March 29th, 2011. The session was moderated by Associate Professor Hans Jrgen Andersen, Department of Arcihtecture, Design, and Media Technology, Aalborg University.

The following adjudication committee was appointed to evaluate the thesis. Note that the supervisor was a non-voting member of the committee.

Reader Ian Reid, Ph.D.
Department of Engineering Science
University of Oxford
Oxford, United Kingdom

Assistant Professor Vittorio Ferrari, Ph.D.
Department of Information Technology and Electrical Engineering
ETH Zurich
Zurich, Switzerland

Associate professor Claus B. Madsen, Ph.D. (committee chairman)
Department of Architecture, Design, and Media Technology
Aalborg University
Aalborg, Denmark

Thomas B. Moeslund, Ph.D. (supervisor)
Computer Vision and Media Technology Laboratory
Department of Arcihtecture, Design, and Media Technology
Aalborg University
Aalborg, Denmark

Abstract

The human motion contains valuable information in many situations and people frequently perform an unconscious analysis of the motion of other people to understand their actions, intentions, and state of mind. An automatic analysis of human motion will facilitate many applications and thus has received great interest from both industry and research communities.

The focus of this thesis is on video-based analysis of human motion and the thesis presents work within three overall topics, namely foreground segmentation, action recognition, and human pose estimation.

Foreground segmentation is often the first important step in the analysis of human motion. By separating foreground from background the subsequent analysis can be focused and efficient. This thesis presents a robust background subtraction method that can be initialized with foreground objects in the scene and is capable of handling foreground camouflage, shadows, and moving backgrounds. The method continuously updates the background model to maintain high quality segmentation over long periods of time.

Within action recognition the thesis presents work on both recognition of arm gestures and gait types. A key-frame based approach is presented to recognize arm gestures. The method extracts a set of characteristic poses and describes them by their local motion resulting in motion primitives. A probabilistic edit distance is used to classify a sequence of motion primitives as a gesture. This 2D recognition process is extended into a view-invariant recognition of arm gestures by use of a range camera that generates 3D data and allows for a 3D equivalent of motion primitives. The recognition of gait types takes a different approach and extracts silhouettes that are matched against a database. A gait continuum is introduced to better describe the whole range of gait which deals with an inherent ambiguity of gait types.

Human pose estimation does not target a specific action but is considered as a good basis for the recognition of any action. The pose estimation work presented in this thesis is mainly concerned with the problems of interacting people and the complex occlusions that interactions produce. A pose estimation method based on the pictorial structures framework is presented. Body part detection combines edge and appearance information in a dynamic way. Occluded body parts are detected by pruning the foreground mask into a mask of possible occlusions. A multi-view approach to pose estimation is also presented that integrates low level information from different cameras to generate better pose estimates during heavy occlusions.

The works presented in this thesis contribute in these different areas of video-based analysis of human motion and altogether bring the solution of fully automatic analysis and understanding of human motion closer.

Resume

Menneskers bevægelser indeholder meget information, som kan bruges i mange situationer, og vi laver hele tiden en ubevidst analyse af andres bevægelser for at forstå hvad de gør, hvad deres intention er og hvordan deres sindstilstand er. Ved at automatisere denne form for analyse åbnes der mange nye anvendelsesmuligheder, og det er derfor et meget interesant problem for både industrien og forskningsmiljøerne.

Fokus for denne afhandling er analyse af menneskers bevægelser ved hjælp af video og afhandlingen præsenterer arbjede indefor tre overordnede områder, nemlig forgrunds segmentering, genkendelse af bevægelser og estimering af kropskonfigurationer.

Forgrunds segmentering er ofte første vigtige skridt mod analyse af menneskers bevægelser. Ved at adskille forgrund og baggrund kan den efterfølgende analyse gøres mere fokuseret og effektiv. Denne afhandling præsenterer en robust metode til baggrunds subtraktion. Metoden kan initialiseres selvom der er forgrundsobjekter i billedet og metoden håndterer forgrundskamuflage, skygger og bevægelser i baggrunden. Metoden opdaterer kontinuert modellen for baggrunden for at opretholde en høj kvalitet af segmentering over lange perioder.

Indenfor genkendelse af bevægelser indeholder afhandlingen beskrivelser af både genkendelse af armbevægelser og genkendelse af gangarter. Til genkendelse af armbevægelser præsenteres en metode baseret på key-frame princippet, hvor karakteristiske armkonfigurationer beskrives ved hjælp af deres lokale bevægelse, såkaldte bevægelsesenheder. En sekvens af bevægelsesenheder kan klassifiseres som en armbevægelse ved hjælp af Edit Distance algoritmen i en sandsynlighedsbaseret udgave. Denne metode udvides til en tredimensionel metode ved hjælp af et time-of-flight kamera, der genererer 3D data. Disse 3D-bevægelsesenheder gør det muligt af genkende armbevægelser uanset kameravinklen. Genkendelsen af gangarter anvender en anden metode der ekstraherer silouetter og sammenligner disse med indholdet af en database. For bedre at kunne beskrive alle gangarter introduceres et kontinuum for gangarter, der håndterer den tvetydighed som eksistrerer ved gangarter.

Estimering af kropskonfigurationer er ikke rettet mod genkendelse af specifikke bevægelser, men kan betragtes som et godt fundament for genkendelse af alle bevægelser. Estimeringen af kropskonfigurationer i denne afhandling er hovedsageligt rettet mod de problemer der opstår når personer interagerer og når de skygger for hinanden på komplekse måder. Estimeringen af kropskonfigurationer bygger på Pictorials Structures metoden. Kropsdele blive detekteret ved en dynamisk kombination af kantinformation og information om deres udseende, og kropsdele der ikke er synlige bliver undersøgt ved at indsnævre forgrundsarealet til kun at dække områder hvor usynlige kropsdele kan forekomme. Estimering af kropskonfigurationer ved hjælp af flere kameraer bliver også beskrevet. Denne metode integrerer information fra flere kameraer tidligt i processen for derved at kunne generere bedre estimater af kropskonfigurationer når store dele af kroppen ikke er synlig.

Det arbejder der præsenteres i afhandlingen bidrager til hvert af disse områder af analyse af menneskers bevægelse, og samlet set briger afhandlingen en fuldautomatisk analyse tættere på realisering.

Preface

This thesis documents my main research activities from 2005 to 2010. I received my M.Sc.EE in June 2005 and have been working as a research assistant at the Laboratory of Computer Vision and Media Technology, Aalborg University since September 2005.

The work has been funded by different projects giving the thesis four different focus points within automatic video-based analysis of human motion, namely foreground segmentation, recognition of arm gestures, recognition of gait types, and human pose estimation. The thesis consists of a collection of published texts together with an introduction that provides an overview of the topic, reviews the publications, and highlights the main contributions.

The following projects have funded the work presented in this thesis:

MoPrim: Motion primitives for a communicative human body language, Danish Research Council project, 2004-2007

HERMES: Human-Expressive Representations of Motion and Their Evaluation in Sequences, EU project (FP6 IST-027110), 2006-2009

BigBrother: Big Brother is watching you!, Danish Agency for Science, Technology, and Innovation, 2007-2010

I have met and cooperated with many highly qualified and inspiring people during the past five years who have influenced my work and motivated me. First of all I would like to thank Thomas Moeslund for his guidance throughout this period and for initiating and supporting the process of writing this thesis. I would also like to thank Michael B. Holte and Serhan Cosar who have been my closest collaborators at different times.

Finally I would like to thank my family. *En stor tak til min familie, som har været en uvurderlig hjælp når arbejdet har krævet en ekstra indsats. Tak Heidi, for din støtte, din tålmodighed og din tro på mig. Tak Maja og Cecilie, for at hjælpe mig til at se tingene i det store perspektiv.*

Preben Fihl

Aalborg, January 2011

Contents

Preface **i**

1 Introduction **1**
1.1 The focus of this thesis . . . 2
1.2 Overview of this thesis . . . 3
1.3 Contributions . . . 17
1.4 Data sets for action recognition . . . 18
1.5 Publications of the thesis . . . 19
References . . . 21

2 Foreground Segmentation **25**
2.1 Introduction . . . 28
2.2 Figure-ground segmentation . . . 28
2.2.1 Background representation . . . 28
2.2.2 Background initialization . . . 29
2.2.3 Background updating . . . 30
2.2.4 Bounding box representation . . . 31
2.3 Tracking . . . 32
2.3.1 Representation . . . 32
2.3.2 Matching . . . 33
2.4 Results . . . 34
2.4.1 Test of figure-ground segmentation . . . 34
2.4.2 Test of tracking . . . 35
2.5 Conclusion . . . 36
References . . . 36

3 2D Human Gesture Recognition **39**
3.1 Introduction . . . 42
3.2 Paper content and system design . . . 42
3.3 Feature extraction . . . 43
3.4 Recognition of primitives . . . 46
3.4.1 Learning models for the primitives . . . 47
3.4.2 Defining the primitives . . . 48

3.5 Recognition of actions . . . 49
3.6 Results . . . 51
3.6.1 Test setup . . . 51
3.6.2 Tests . . . 52
3.7 Conclusion . . . 52
References . . . 53

4 3D Human Gesture Recognition 57
4.1 Introduction . . . 60
4.1.1 Our approach . . . 61
4.1.2 Structure of the paper . . . 62
4.2 Segmentation . . . 62
4.2.1 Data acquisition and preprocessing . . . 62
4.2.2 3D motion detection . . . 63
4.3 Motion primitives . . . 65
4.3.1 Motion context . . . 65
4.3.2 View-invariant representation: harmonic motion context . . . 68
4.4 Classification . . . 70
4.4.1 Recognition of primitives: correlation . . . 70
4.4.2 Recognition of gestures: probabilistic edit distance . . . 71
4.5 Test and results . . . 73
4.5.1 Unknown start and end time . . . 75
4.6 Conclusion . . . 76
References . . . 76

5 Gait Type Analysis 81
5.1 Introduction . . . 84
5.1.1 Gait type description based on the Duty-factor . . . 85
5.2 The Duty-Factor . . . 87
5.3 Silhouette extraction . . . 90
5.4 Silhouette description . . . 90
5.5 Silhouette database . . . 91
5.6 Silhouette comparison . . . 93
5.7 Gait analysis . . . 94
5.7.1 Action error . . . 95
5.7.2 Action likelihood . . . 95
5.7.3 Temporal consistency . . . 97
5.7.4 Gait-type classification . . . 97

5.7.5 Duty-Factor calculation . . . 97
5.8 Results . . . 99
5.8.1 Gait-type classification . . . 99
5.8.2 Duty-factor . . . 101
5.9 Discussion . . . 103
5.10 Multi camera setup . . . 104
5.11 Real time performance . . . 106
5.12 Online parameter tuning of segmentation . . . 106
5.13 Conclusion . . . 108
5.14 Acknowledgment . . . 109
References . . . 109

6 2D Human Pose Estimation 113
6.1 Introduction . . . 116
6.2 Pictorial structures framework . . . 117
6.3 Foreground segmentation . . . 118
6.4 Dynamic optimization of appearance model . . . 119
6.5 Occlusion handling . . . 120
6.6 Results . . . 124
6.7 Discussion . . . 127
References . . . 127

7 Multi-view Human Pose Estimation 129
7.1 Introduction . . . 132
7.2 Single-view pictorial structures framework . . . 133
7.3 Multi-view pose integration . . . 134
7.4 Results . . . 137
7.5 Discussion . . . 137
7.6 Conclusion . . . 139
References . . . 139

8 Conclusion 143
8.1 Discussion and outlook . . . 144

A Data set listings 147

Chapter 1

Introduction

The analysis of human motion provides valuable information in hundreds of situations. People are doing such analysis every day to find out where other people are, where they are coming from or heading to, how they are moving, and what or who they are interacting with. Through this analysis we are able to identify friendly, suspicious, or threatening behavior, identify social relations among people, get a first impression of the emotional state of other people, and react appropriately to all this information.

This very general problem of analyzing human motion has been a main interest of many research communities for years with the goal of replicating the human ability to understand human motion. A successful analysis of human motion facilitates numerous applications within a broad range of research and industrial fields. Psychologists have investigated the correlation between how people move and their emotional state as a diagnostic tool. Neuroscientists have investigated how people infer the intension of human motion to understand the cognitive processes of human behavior. Within clinical diagnostics human motion is analyzed to diagnose a number of diseases. In sports human motion is analyzed to optimize performance. The entertainment industry is producing motion analysis tools to allow for new interaction methods with game consoles and to capture human motion for computer graphics special effects. Law enforcement, intelligence agencies, governments, and the security industry in general are analyzing human motion in surveillance videos to identify illegal or suspicious behavior or to identify individuals by the way they move. Media providers are looking for ways to automatically understand and annotate video to make it searchable and add a commentary. And within robotics the analysis of human motion will allow for robots that can work alongside humans and interact with humans in a natural way.

In these applications the analysis of human motion will most often be done using recorded video instead of just observing people. For some applications the videos are captured specifically with human motion analysis in mind. For example, for

diagnostic analysis and performance optimization in sports this allows physicians and coaches to conduct a much more thorough analysis. The video can be viewed in slow motion, it can be replayed many times, and synchronized video from multiple viewpoints can be recorded. For human-computer interaction the video is also captured with human motion analysis in mind and with people knowingly conducting certain motions but the surroundings and the way the motion is conducted are rarely controllable leading to a more difficult analysis. For surveillance applications, video annotation, and robotics applications the video is most often captured in unconstrained environments with people doing real-life motions rather than instructed performances of specific motions. Such unconstrained video makes the analysis of human motion a very challenging task.

To reach the true potential of all these applications the analysis of human motion needs to be completely automated and conducted in a robust manner. This has been a prime goal of computer vision research for a long time and human motion analysis is still the focus of huge amounts of research that addresses many different aspects of this problem. One of the main driving applications has been automated analysis of surveillance video which has been partly motivated by the great political focus on security and prevention of terrorist attacks in recent years. This type of video analysis requires methods that efficiently deal with very challenging lighting conditions, diverse scenes, and unconstrained human motion, interaction, and grouping.

The computer vision research presented in this thesis does not target a specific application of human motion analysis but rather presents work on systems that can enable many different applications.

The focus of this thesis will be specified next (section 1.1) and the reminder of this chapter will after that be structured as follows. Section 1.2 will elaborate on the focus of the thesis by presenting an outline of each of the chapters and state how the different methods and chapters relate to each other. Section 1.4 introduces a comparative data set listing presented in appendix A. Section 1.3 will highlight the main contributions of this thesis and section 1.5 lists all the publications that have been published in relation to the work of this thesis.

1.1 The focus of this thesis

The work of this thesis deals with video-based analysis of human motion and how this analysis can facilitate an understanding of the actions that people perform.

Following the taxonomy of [19] human motion analysis centers around the processes of *tracking*, *pose estimation* and *recognition*, all relying on a proper *initialization*.

Tracking is the process of segmenting a person from the background and finding correspondences between the segmentations of consecutive frames. In tracking the human body is most often considered to be a rigid object or an articulated rigid object and analyzing the motion of this object provides information about where people are moving, how fast they move, and the general motion patterns of a scene.

This thesis includes work on robust segmentation that can facilitate tracking but most importantly in the context of this thesis enables the localization of a person and provides the corresponding silhouette. A method for establishing correspondences between segmented objects in consecutive frames is also presented but this is not a main focus of the thesis.

Pose estimation addresses the problem of finding the pose of the individual body parts and recognition is both recognizing individuals in a video and their actions and activities. Pose estimation and recognition of actions is the main focus of this thesis. The terms actions and activities are used differently in the literature but here actions refer to a short sequence of poses or body configurations that can be recognized as an intentional act of a person. The term gesture is used as a synonym for action, mainly when describing arm actions. Activities are larger scale events that describe sequences of actions in a certain context. An example of these terms could be kicking and running as actions and playing football as an activity. The actions kicking and running can be recognized from a short sequence of body configurations (although the execution of the running action may last for a long time). The activity "playing football" consists of for example running on a football field and kicking a ball. In a different context the same actions can represent another activity, for example, running in the street and kicking parked cars can be one representation of the activity "vandalism".

Doing segmentation, pose estimation, and action recognition in laboratory environments or carefully designed scenes can provide valuable information about the performance of methods and systems. However, the variability and challenges of real-life scenes are not investigated in this way, often leading to less general and less applicable methods. The work of this thesis mainly focuses on scenes that are not carefully constrained. The work on segmentation specifically targets dynamic outdoor scenes and the work on pose estimation builds on this and furthermore addresses unconstrained motion of multiple people. One part of the work on action recognition is applied in an office environment whereas the other part addresses the challenges of dynamic outdoor scenes and changing motions paths.

Human motion analysis is a very active field of research which for example can be seen by the number of publications reviewed in recent surveys [12, 19]. The amount of publications within human motion analysis results in surveys and reviews that focus on specific areas within human motion analysis rather that all aspects of the domain, for example, [1] focus on body modeling and recognition of actions and activities, [9] reviews detection and tracking of people, [13] present a review focusing purely on view-invariant pose estimation and action recognition, [21] reviews pose estimation, and [22] reviews action recognition.

1.2 Overview of this thesis

This thesis consists of eight chapters with the current chapter being the first. The following six chapters each consists of a previously published text. Each chapter has

a brief introduction explaining the context of the publication. The eighth chapter concludes on the thesis. The following will give an overview of each of the chapters. The relation to human motion analysis and the other chapters will be presented together with an outline of the methods used and a summary of the results.

Chapter 2. Foreground Segmentation

This chapter presents work on foreground segmentation and tracking. To localize the foreground objects in a scene, here being people in the scene, is in many methods the first step in the analysis of human motion. For a number of applications, like tracking, it is sufficient to extract the location and scale of a person and many methods have been presented to detect people in images, *e.g.* [7], [8], and [15] (see [9] for a recent survey). However, the extraction of the silhouette of a person is needed in many pose estimation and action recognition applications and background subtraction methods are applied in many systems for segmentation.

For background subtraction both pixel-based methods and region-based methods have been proposed. Pixel-based methods classify each pixel as foreground or background individually whereas region-based approaches classify a pixel based on the region surrounding it or alternatively classify the whole region in one step. Both types of background subtraction can incorporate temporal information to improve segmentation. In general terms, region-based approaches produce better segmentation results but at the expense of increased processing demands. However, when looking at background subtraction as a preprocessing step to human motion analysis a fast pixel-based method is often preferred to allow more time consuming processing in the later steps. The pixel-based methods may produce more noisy segmentations but rather simple filtering can significantly reduce the amount of noise and perfect segmentations are rarely needed in the subsequent processing steps of the human motion analysis.

To enable human motion analysis in diverse and realistic environments a robust background subtraction method is required that can maintain segmentation quality over long periods of time. Chapter 2 presents a pixel-based background subtraction method that addresses the challenges of dynamic outdoor scenes and updates the background model to allow for hours of continuous segmentation.

The background subtraction method is shown to be robust in handling foreground camouflage, shadows and illumination changes. This is achieved by a separation of the chromatic and intensity information in the background subtraction step. Many background subtraction methods achieve this by using different color spaces than RGB, for example HSV or normalized RGB, to isolate the chromatic changes of pixels from changes in intensity. In the approach presented in chapter 2 colors are represented in RGB but separate thresholds are calculated for chromaticity and intensity. This makes the background subtraction capable of allowing relatively large variations in intensity, typically due to shadows and gradual changes in illumination levels, while keeping a good separation of foreground objects with colors close to

those in the background (foreground camouflage).

The background model is represented by a set of codewords for each pixel. Each codeword contains information on the mean RGB-color, minimum and maximum allowable intensities, and temporal information about when and how often the codeword is observed. The codewords are created during a training phase and by allowing multiple codewords for each pixel it is possible to model backgrounds with alternating appearances. Tree branches that wave in the wind are an example of a background that is modeled in this way. This is a key feature of the background model in order to achieve robust segmentation in outdoor scenes that do not have completely static backgrounds.

Another important property of the background model is the ability to initialize without having a completely empty scene. For busy public spaces it can be very difficult to ensure a training period without any foreground objects so the presented method utilizes the temporal information of the codewords to distinguish between true background and foreground objects that pass by. The distinction is made based on the *m*aximum negative run-length (MNRL) which express the longest time between two observations of a codeword during a giving training period. True background, which includes for example moving tree branches, will be observed quasi-periodically resulting in a short MNRL. Foreground objects will be observed for a short period of time and then be gone for a long period of time resulting in a long MNRL even though the number of observations may be the same as for moving backgrounds.

The background subtraction method has two different update mechanisms to handle gradual and rapid changes respectively. To handle gradual changes the codewords are continuously updated by taking a weighted average of the values of the observed codeword and the new pixel values. To avoid updating based on falsely classified background the continuous update process is only applied to stable background, *i.e.* pixels that have been classified as background for a short period of time. This update procedure does not handle rapid changes in the background and a different update process is therefore also applied. By allowing new codewords to be included in the background model at runtime it is possible to adapt to new background after a short training period. If cars are parked in the scene or the weather changes the appearance of parts of the scene (*e.g.* by wind or rain), then new codewords will be observed with a short MNRL during some training period which means that they represent true background and not foreground objects.

To analyze the segmentation results in the context of human motion analysis chapter 2 also presents an appearance based tracking method. The segmented foreground regions are classified as people or noise by fitting an ellipse to the region and then evaluating a set of criteria based on size, proportions, direction of major axis, and fidelity of the fit between the ellipse and the foreground region. Correspondences between individuals from one frame to the next are then established using an appearance model to associate the identities of foreground regions. Each person is represented by a feature vector describing the image location, the mean color of the leg region, and the mean color of the upper body region. Colors are represented

as HSV for this purpose and only the hue and saturation components are used in the feature vector. The similarities between all extracted persons in one frame and all extracted persons from the previous frame are calculated using the Mahalanobis distance and associations are established in a greedy manner. An entry/exit zone is defined near the image borders to ensure that the identities of people that enter or leave the scene are not associated to people in the scene.

Chapter 2 evaluates both the segmentation results and tracking results in a 10 hour long outdoor video captured from 9.15am to 7.15pm. For testing the background subtraction a set of frames is sampled from the whole time span of the video. 93 frames containing people are sampled and the foreground regions are manually annotated. These frames are used to calculate the false rejection rate (FRR). For the calculation of the false acceptance rate (FAR) an additional 971 frames are sampled containing only background (not needing manual annotation). The most important result of this test is that the performance of the background subtraction method does not decrease significantly over the course of day. Challenging situations like strong gusts of wind and low illumination level at dusk cause relatively high false rejection rates and false acceptance rates but this is not dependant on how long the background subtraction have been running and the background model adapts to the changing situations. For testing of the tracking the test set consists of 267 people moving through the scenes during the 10 hours of video. The tracking failed in 35 cases, corresponding to 13.1%, but when at looking the tracking results for each 30 minutes interval it is clear that the performance of the tracking maintains stable throughout the day and tracking errors are mostly occurring when the foreground segmentation is noisy.

The background subtraction algorithm is used as a pre processing step in chapters 5 and 6 where the robustness of the method makes it possible to also do action recognition and pose estimation in realistic outdoor environments.

Chapter 3. 2D Human Gesture Recognition

This first chapter on action recognition investigates a key-frame approach for the recognition of arm gestures. The gestures are five one-arm gestures used in the communication between people over longer distances. The gestures are: point right, move right, move left, move closer, and raise arm.

Many recent methods recognize actions directly on image data rather than extracting joint locations and then recognize actions based on this representation. Action recognition from silhouette data is an example of this which is also the approach taken to action recognition in chapter 5. Another very popular approach in recent years has been recognition using spatio-temporal interest points or spatio-temporal volumes that both base the recognition on information from all frames of a sequence. In chapter 3 an approach is presented that base the action recognition on a small set of characteristic poses (key-frames) that can be reliably detected. Key-frame approaches base the recognition on a subset of the entire sequence with the assumption

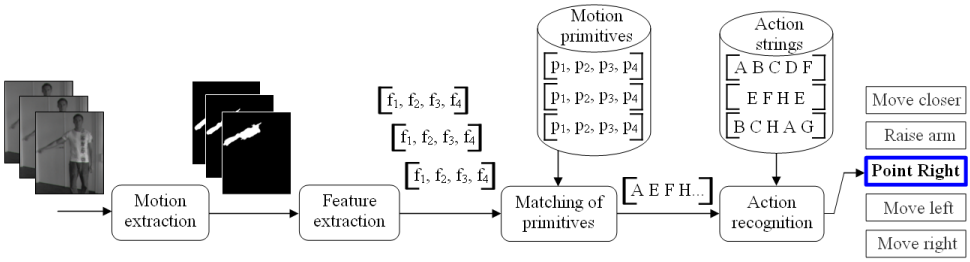


Figure 1.1: An overview of the approach for recognition of human gestures. Motion is extracted and represented with a four-dimensional feature vector. The feature vectors are matched to a set of trained motion primitives. This results in a sequence of primitives representing the action performed in the input video. The action recognition step compares an incoming sequence of primitives with a set of trained action models and classifies the action.

that certain characteristics of an action are more easily recognized than others and basing the recognition on the frames containing these characteristics yields a more robust result. A recent and extensive survey on action recognition is given in [22].

The approach of chapter 3 finds the characteristic arm poses based on the arm motion around each pose. The representation used for these poses are denoted *m*otion primitives. Figure 1.1 gives an overview of the whole approach.

Motion is detected using double differencing, *i.e.* using three consecutive frames to generate two difference images which are thresholded and combined by a pixel-wise logical AND. The thresholding utilizes a hysteresis principle to eliminate noise. The result of the motion detection is a binary blob describing the motion of the arm.

To extract a set of features for the motion blob it is modeled by an ellipse and four scale invariant features are calculated. The features describe the shape, orientation, and location of the ellipse with the location defined relative to a reference point on the person. Based on the extracted feature vector each incoming frame will be classified as either belonging to one of the motion primitives or as a noise class.

The representation of the motion primitives is based on a set of training samples for each pose. To acquire the training data magnetic trackers are placed at the joints of the arm on training subjects. Each training subject repeats all arm gestures and the trajectories of the tracker markers are transferred to a computer graphics model of a person. The animations of the graphics model constitute the training data for the approach. This semi-synthetic training data (*i.e.* gestures performed by humans but synthesized with a graphics model) decouples the training data from the image data used in the recognition process. For other approaches, like methods based on spatio-temporal interest points, the training and testing data are typically different but with the exact same image characteristics. The motion based primitives and the semi-synthetic training data make the approach presented in chapter 3 applicable to more diverse input data.

The subsequences defining the motion primitives (three frames for each primitive) are found manually. The criteria for selection of the subsequences are the following: Firstly, that the subsequence represents a characteristic and representative 3D configuration. Secondly, that a certain amount of motion is present in the subsequence. Thirdly, that the subsequence is representative for as many gestures as possible. The third criteria results in a small set of 10 robust primitives for the five actions with each action represented by five to eight of these primitives.

The set of semi-synthetic subsequences that represent the training data for a motion primitive is processed to find the feature vectors and each primitive is represented by the mean and covariance for these vectors. The feature vectors of testing video can now be classified using the Mahalanobis distance. If the minimum Mahalanobis distance is above a certain threshold then the feature vector is classified as noise. For a test video this classification will result in a sequence of primitives representing the gesture being performed in the video.

The classification of a sequence of primitives as one of the five gestures is done using a novel extension of the edit distance that incorporates the likelihood of each primitive. The original edit distance express the number of operations needed to convert one sequence into another where possible operations are insertion of a symbol, deletion of a symbol, or exchange of a symbol with one from the other sequence. Each operation can have an associated cost which in chapter 3 is extended to a cost dependant on the probability of each observed symbol, here being each motion primitive. The probability of a motion primitive is expressed through the number of observations of a given primitive. The edit distance is furthermore normalized with the length of the sequence of primitives to avoid bias towards short sequences.

To test the method a set of 550 video sequences is captured, each containing the execution of a gesture. Two different test setups are used. In the first setup each test video contains the execution of one gesture. This could imitate gesturing for human-computer interaction where it is known when commands are being issued. The second test setup imitates the more realistic problem of not knowing when the execution of the gesture commences and when it terminates. This is achieved by adding executions of half of a gesture to the start and end of the original captured video sequences. The gestures used for these half executions are chosen randomly. The overall recognition rates for the two test setups respectively are 88.7% and 85.5%. Most of the erroneous classifications are a result of confusion between the *m*ove closer and *r*aise arm gestures. Seen from a fronto-parallel view the motion of the two gestures are somewhat alike. This issue is one of the motivating factors for extending the work of chapter 3 into a view invariant representation which is the presented in chapter 4.

Chapter 4. 3D Human Gesture Recognition

The approach of chapter 3 is in this chapter extended into a view invariant gesture recognition method based on 3D input data. A time-of-flight range camera is used

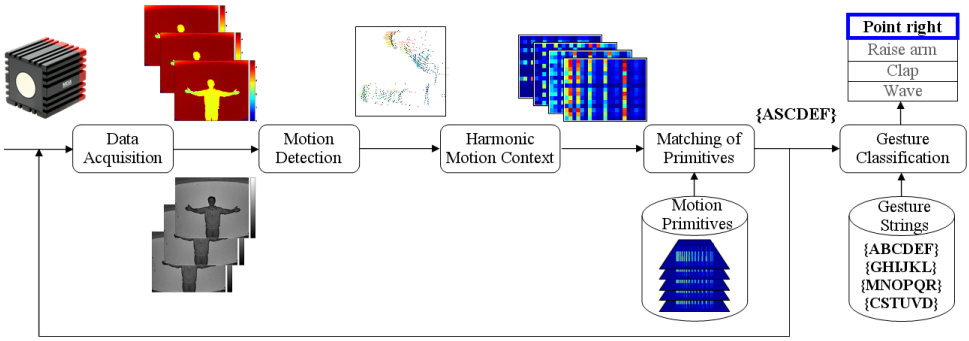


Figure 1.2: A schematic overview of the 3D human gesture recognition. Recognition is based on range and intensity images. 3D motion is extracted and described by harmonic motion contexts. These are matched to trained motion primitives. A number of frames are processed this way (illustrated by the feed back loop) resulting in a sequence of primitives. This sequence is classified against a set a trained gesture strings.

to produce both a depth map and an intensity image which allow for the extraction of motion in 3D. The time-of-flight camera ensures a direct alignment of the depth and intensity information as opposed to the classical stereo approaches which have to be carefully calibrated and establish correspondences between cameras.

Another important difference between the approach presented here and other related methods is the characteristics of the training data. When addressing invariance to viewpoint in gesture recognition the training data often includes video captured from different viewpoints. A view-invariant representation of the actions ensures that a test video can be classified without first recovering the viewpoint of that sequence (see for example [26] and [6]). The approach presented in chapter 4 reduces the training data to a single viewpoint while maintaining the ability to recognize gestures from different viewpoints, say viewpoints rotated ± 45 degrees from the training viewpoint.

The recognition is based on the notion of motion primitives. Here, the motion primitives describe both the amount of motion and the 3D direction of the motion (as opposed to the binary motion detection of chapter 3). Figure 1.2 shows an overview of the method. The motion is detected using a 3D version of optical flow. The detected motion is represented using motion contexts (an extension of Shape Contexts). The representation is made invariant to rotation around the vertical axis using spherical harmonic basis functions, yielding a harmonic motion context representation.

In each frame the motion primitive which best explains the observed data is found. This is done by calculating the normalized correlation coefficients between the harmonic motion contexts of the observed data and the motion primitives. A video sequence will in this way result in a sequence of primitives representing the action

performed in the video. The classification of a sequence of primitives is done by use of the probabilistic edit distance of chapter 3.

The method is used to recognize four one- and two arms gestures, namely "point right", "raise arm", "clap", and "wave". These four actions are represented using 22 motion primitives. The method is tested on 160 video sequences. The sequences show 10 test subjects performing two repetitions of each gesture. The gestures are captured from two viewpoints, one frontal view and one view rotated 45 degrees. As stated above, only data from the frontal view is used for training while the testing includes both viewpoints. The test uses the same test protocol as in the 2D gesture recognition system where one test is conducted with exactly one gesture per sequence (known start and end times) and one test has "noise" gestures (half executions of gestures) added to the beginning and end of the sequences (unknown start and end times). The method achieves a recognition rate of 94.4% when the start and end times are known and a recognition rate of 86.9% when the start and end times are unknown.

Chapter 5. Gait Type Analysis

This chapter presents work on the recognition of gait actions, i.e. the three main gait types walking, jogging, and running. In the literature, gait often refers to the style of walking which can be used to identify individuals. In this thesis however, gait describes the different types of the bipedal human motion. The action recognition problem addressed in chapter 5 focuses on unconstrained environments. As stated earlier, real-life outdoor scenes impose a set of challenges on methods for video-based analysis of human motion, challenges like changing illumination conditions, dynamic backgrounds, people observed at different scales and from different viewpoints, and people moving in an unconstrained manner (*e.g.* changing direction or speed of motion). The robustness of action recognition towards these types of challenges has to some extent been addressed by other methods as well, but the methods combined in this gait type classification system are chosen and developed with robustness in mind making the system capable of handling more of these factors than comparable approaches.

The most important difference between the system presented in chapter 5 and other state-of-the-art methods is the use of a unified continuum to describe gait types as opposed to the traditional notion of three distinct gait types. The gait continuum introduced here is based on a well established physical property of gait, the duty-factor, described in the biomechanics literature. The duty-factor describes for each foot the fraction of a gait cycle where that foot is on the ground. The main benefit of using the duty-factor to describe gait is for representation of jogging and running. A clear definition of the difference between jogging and running is not easily given and when several people are asked to classify gait actions as either jogging or running it results in an ambiguous classification for certain executions. The duty-factor representation allows jogging and running to be described as a point on a continuum instead of either one type or the other. By including walking in this gait continuum

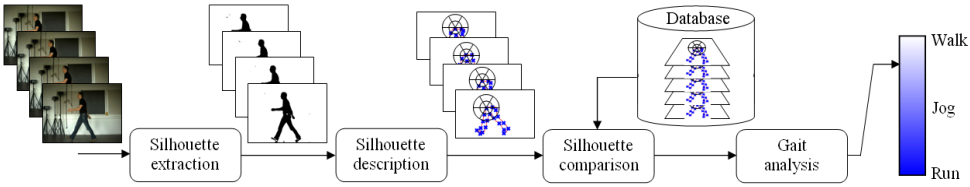


Figure 1.3: An overview of the approach for gait type analysis. Silhouettes are extracted and represented by shape contexts. The silhouettes are then matched to a database of synthetically generated silhouettes. The gait analysis collects a sequence of matches and estimates the duty-factor from this sequence.

it is possible to describe all gait types using just one parameter. It should be noted that the approach in chapter 5 first performs a classification of gait into three main types as an intermediate step and then apply the duty-factor representation. The classification of jogging and running as types is in terms of the duty-factor just partly overlapping classes.

The approach utilizes a computer graphics model of a person to generate training data like in chapter 3 but here the motion of the model is also synthetic. This results in a complete separation of training and testing data. The motion of the computer graphics model constitutes just one gait cycle of each gait type and is created simply by use of a tool for animation of walking and running. With the pure computer graphics representation of the training data it is very easy to generate training data from different viewpoints and viewpoints rotated ± 30 degrees from a direct side-view are included in the training set. This generic execution of the three gait types is enough to capture the main properties of each action resulting in a very small training set compared to other approaches. Altogether, only 270 frames are used for training. A main reason why this small training set works for recognition is because of the actions in question. Fairly subtle differences between gait patterns of different people can be found that allow for identification of individuals but the main components of the motion in gait are somewhat universal and can therefore be modeled by one prototypical execution. Other real-life actions may not have the same universal representation (as opposed to the choreographed actions of many data sets that seem to be performed according to one prototypical execution). It is interesting to note that the cyclic nature of gait is not a prerequisite for the recognition based on the synthetic training data. The gait type classification can be performed on just two gait cycles with high rates of correct classification.

The method presented in chapter 5 is based on matching of silhouettes with a database containing the training data. Figure 1.3 provides a schematic overview of the whole approach. Silhouettes are extracted using the background subtraction method of chapter 2. The gait classification is based purely on the silhouette of the legs disregarding any motion of the arms which potentially could be other non-related actions like waving or pointing. The leg silhouette is simply defined as the bottom half of the whole silhouette.

The extracted silhouettes are represented using shape contexts, *i.e.* log-polar histograms for points sampled around the contour of the silhouettes and orientations of the tangent of the contour at those points. This representation is scale and translation invariant. [16] present an enhancement of the shape contexts that use the silhouette's inner distance between the sampled points which is shown to give improved shape matching results. The advantage is however mainly valid for articulated objects and does not produce significantly better results for the rather simple leg silhouettes of the presented approach.

The shape context representation of the silhouettes are matched to the silhouettes of the database using the χ^2 test statistics and the Hungarian distance. The χ^2 test statistics is just one out of many methods to calculate the distance between the log-polar histograms. The Earth Mover's distance [23] or the quadratic-chi histogram distance [20] that take cross-bin relations into account often give more accurate results but the effect is negligible when sampling the contour densely. The χ^2 test statistics is on the other hand calculated very efficiently in the present implementation. The output of the silhouette matching is a dissimilarity measure between the test silhouette and each of the silhouettes in the database.

When a sequence of frames has been processed by the silhouette matching the gait analysis is conducted in two steps. The sequence contains at least a single gait cycle. First, the sequence is classified as belonging to one of the three main gait types. Second, the duty-factor is calculated for the sequence.

The classification of gait type is done by a combination of three different types of information. First, an *action error* is calculated based on the distance from the silhouettes of the sequence to the best matching silhouettes of the database. The action error is calculated for each gait type individually. Next, two sets of weights are calculated that are applied to the action error. The first weight is the *action likelihood* that express the percentage of frames for which the single best matching database silhouette belongs to a given gait type. The second weight is *temporal consistency* which express the percentage of consecutive frames where the single best matching database silhouette belongs to a given gait type. The two sets of weights are multiplied with the action errors and the gait type of the test sequence is the gait type with the smallest weighted error. Only the database silhouettes of this gait type are used for the calculation of the duty-factor.

The silhouettes of the database have been annotated with the number feet touching the ground. By estimating the number of frames in a gait cycle the duty-factor can then be calculated from these annotations. The gait cycles are analyzed using the width of the silhouette which oscillates between a small width when the feet are close together and a large width when the feet are furthest apart. The number of frames between two peaks or two valleys corresponds to the number of frames in half a gait cycle.

The method is tested on a set of 138 diverse sequences compiled from four different datasets. The test data contains indoor and outdoor video, different moving directions with respect to the camera (up to ± 45 degrees from the viewing direc-

tion), non-linear paths, different camera elevations and tilt angles, different video resolutions, and varying silhouette heights. The gait type classification is tested on its own to allow a comparison with other state-of-the-art action recognition methods. The overall recognition rate is 87.1% and the results are comparable to the recognition rates reported by other methods on similar data. The majority of false classifications are jogging sequences classified as running. The duty-factor estimation is tested against a manual annotation of the duty-factor. The duty-factor is in the range from zero to one and the mean error for the estimation is 0.050.

The chapter also describes an online version of the system that runs in real time and does gait type estimation on a single gait cycle. The most time consuming task in the system is the matching of a test silhouette to all database silhouettes. A significant speed up of this step can be achieved by reducing the number of points sampled from the contour and by reducing the number of bins in the log-polar histograms. Furthermore, only the outermost contour is extracted disregarding holes in the silhouette. These changes result in gait analysis at 12-15 fps with only a small decrease in classification performance.

As stated earlier, the system builds on the segmentation of chapter 2. In chapter 5 an automatic online adjustment of the parameters of the background subtraction is introduced to maintain high quality segmentation in difficult scenarios. An edge-based background model is introduced alongside the codebook background model. With the edge-based background model it is now possible to estimate which edges belong to foreground objects. The parameters of the background subtraction can now be optimized by comparing these foreground edges with the edges of a small set of codebook background segmentations (produced with different parameter setups).

Finally, a fusion of gait type information from multiple cameras is described. Generating database silhouettes with viewpoints rotated ± 30 degrees from a direct side-view allows for gait analysis of people moving at angles of ± 45 degrees with respect to the viewing direction. This constraint can be lifted by combining information from cameras with orthogonal viewing directions. Both the gait type classification and the duty-factor estimation extend easily into a multi-camera setup.

Chapter 6. 2D Human Pose Estimation

Chapter 6 deals with the estimation of the human pose whereas chapters 3, 4, and 5 deal with the recognition of specific actions. The pose estimation problem addressed in this chapter does not target a specific set of actions but is motivated by the idea that a good description of the body configuration at any given time will make it possible to build an action recognition system for any type of action.

The problem of estimating the 2D pose of a person in images or video from a single viewpoint is addressed in many publications [2, 10, 14, 25]. One of the significant challenges for monocular approaches is occlusions, both self occlusions and occlusions by other people or foreground objects. When considering a pose estimation problem with multiple interacting people and with no constrain on the motions they conduct

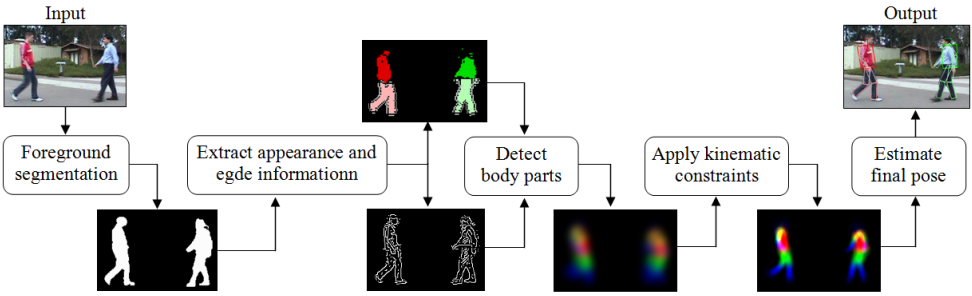


Figure 1.4: An overview of the approach for human pose estimation. First, the foreground segmentation is utilized to reduce the search space. Appearance and edge information is extracted and used in the detection of body parts. The kinematic constraints are applied to the body part detections before the final pose is estimated.

the possible occlusions will be very difficult to model directly and the work presented in chapter 6 therefore takes another approach. The segmentation of chapter 2 is utilized to first localize people which gives a significant reduction in the search space of the subsequent detection of body parts. This could also be achieved by a person detection, like in [10], but using the foreground segmentation makes it possible to additionally search for body parts that are occluded by other people or by foreground objects.

The pose estimation is done using the pictorial structures framework, which has shown good results on human pose estimation and elicited a number of extensions and improvements towards this problem. Body parts are detected using a discriminative color classifier and generic rectangular body part templates. The color classifiers are trained specifically for each person in the scene. The initialization is done on a characteristic walking pose that is easy to detect and where all body parts are typically visible. The body parts detections are combined under a set of pairwise kinematic constraints into a tree-structured body model. Figure 1.4 shows a schematic overview of the approach presented in this thesis.

In the pose estimation process, the body parts are detected using both edge and appearance information. Chapter 6 introduces a dynamic local weighting of the two types of information. The weighting is determined by a local quality measure of the appearance. Different body parts typically have similar appearance (*e.g.* the torso and the arms) and also the appearance of different people can be similar. By estimating the quality of the body part detections based on appearance it is possible to optimize the weighting of edge information versus appearance information for each body part detection individually.

The occlusion handling presented in chapter 6 builds on the initial assumption that everything within the foreground mask that does not have the same appearance as a given body part is potentially occluding that body part. A purely shape-

based detection of body parts is conducted within this region of possible occlusions. The initial assumption is of course not very accurate and the resulting body part detections must therefore comply with a set of extra strong kinematic constraints to be included in the pose estimation.

The method is tested both qualitatively and quantitatively. The tests show the effect of the contributions of chapter 6 by comparing the results of the presented pose estimation with the results of pose estimation without the dynamic weighting and the detection of occluded body parts. The pose estimation for comparison includes the foreground segmentation for search space reduction but not for occlusion handling. It also combines edge and appearance information but does it with a static weighting instead of the presented dynamic local weighting. The qualitative comparison is based on frames from two outdoor video sequences. Both sequences show two people interacting; shaking hands in one sequence and hugging in the other sequence. The comparison shows that the presented approach does improve on the ability to estimate the pose of occluded body parts and body parts that are in front of a region with similar appearance. The quantitative test report results on the ability to correctly localize body parts. 125 frames of the two sequences are used for the test. The ground truth location of each body part has been manually annotated. The percentage of correctly localized body parts increase from 62.9 to 72.9 when the dynamic weighting and the occlusion handling is applied. When specifying results for occluded body parts only the percentage of correctly localized body parts shows a slightly bigger increase, from 40.5% to 54.2%.

Chapter 7. Multi-view Human Pose Estimation

Chapter 6 deals with pose estimation and the problems of occlusions using a single camera. Monocular methods will however always be challenged by heavy occlusions. This chapter will introduce a multi-view approach that extends the work of chapter 6 to more efficiently deal with the problems of heavy occlusion during interaction between people.

Multi-view methods typically belong to one of two categories. Methods in the first category do pose estimation with carefully arranged camera setups in laboratory environments [4, 5, 18]. Methods in the second category work with far less constrained camera setups and environments (typically large open spaces and outdoor scenes) and then deal with tracking instead of pose estimation [11, 17]. The method presented in chapter 7 addresses the problem of pose estimation in outdoor scenes with little constraint on the camera setup.

The presented multi-view approach integrates low level information from multiple cameras to improve on the 2D pose estimates found for each camera. The goal is not to generate 3D pose estimates but rather to improve on the 2D pose estimates from a variable number of cameras with no restrictions on the overall setup. The method uses calibrated cameras and projects data from multiple views into a common 3D space where the probabilities of individual body parts can be combined.

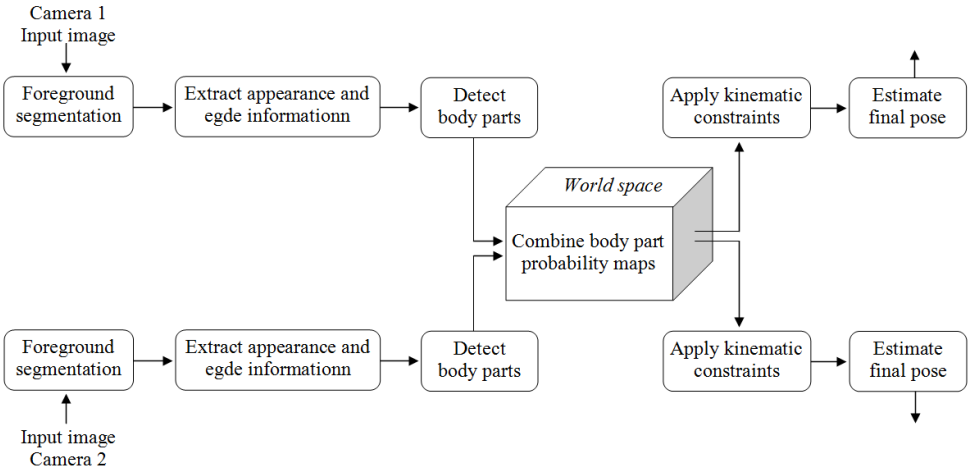


Figure 1.5: A system overview for the multi-view human pose estimation. The approach follows the single-view approach of chapter 6 but combines the result of the body part detections from multiple cameras (here, only two cameras are illustrated). The body part probability maps are projected into world space and the result of the multi-view integration is then projected back to each camera view where the pose estimation is done as in the single-view approach.

The multi-view approach builds directly on the single-view pose estimation of chapter 6. Figure 1.5 provides an overview of the approach. The initialization and detection of body parts are done exactly as in the single-view approach. This results in body part probability maps that represent the probability of a body part at a given image location and with a given rotation. Each body part probability map is converted into two joint probability maps that are independent of the rotation of the body parts. This allows for an efficient integration of probabilities from different views.

All non-zero joint probabilities are now projected into 3D world space. The projection of each probability (each pixel value) represents a line through world space that may intersect the projections from other views. The probabilities are added for such intersections. This projection results in a joint probability volume in 3D world space.

Body parts are now sampled from these joint probability volumes by first clustering the probabilities to get spherical joint representations and next connecting joint clusters corresponding to body parts. Joint clusters are connected if the distance between them corresponds to the appropriate body part, for example connecting a shoulder joint cluster to an elbow joint cluster if the distance between them correspond to the length of an upper arm. The body part samples created in this process are projected back to each camera view where they represent improved body part probability maps. The back-projected body part samples are combined under the same kinematic constraints that are applied in the single-view approach and the rest

of the pose estimation is conducted just like in the single-view case.

Multi-view pose estimation results are presented on two camera views of the PETS 2009 data set. The sequences show two people passing. Pose estimation is done for one person who is fully occluded in the one view and fully visible in the other view. The results show that the presented method can improve pose estimation under full occlusion but also show that the combination of views introduce noise that decrease the performance of the pose estimation when there is no occlusion.

Chapter 8. Conclusion

The conclusion of the thesis will summarize and discuss the main contributions and provide an overview of future research related to segmentation, action recognition, and pose estimation.

1.3 Contributions

The thesis presents work within four overall topics, namely foreground segmentation, gesture recognition, gait type classification, and human pose estimation. This section will point out the main contributions within these areas.

Foreground segmentation in very long video sequences The background subtraction method of chapter 2 introduces a method for continuously updating the background model. This allows the background subtraction method to adapt to the gradual changes in lighting conditions that occur during an entire day. The updating mechanism is tested on a ten hour long video sequence as opposed to the test sequences of a few minutes usually used for testing of background subtraction methods. An approach for online adjustment of the most important parameters of the background subtraction method is furthermore presented in chapter 5.

Motion primitives for action recognition In chapter 3 the simple principle of image differencing is used to extract motion primitives which are represented with a compact four-dimensional feature vector. These simple and compact motion primitives achieves good recognition results on arm gestures but are not limited to this type of actions. The motion primitives are a principled representation that can be used for a wide variety of actions. The action recognition by probabilistic edit distance can also easily be used to classify other actions that are represented by a set of primitives.

Synthetic training data for action recognition The use of a computer graphics model of a person for generation of training data in action recognition methods is explored in both chapter 3 and chapter 5. In the first case real human motion is captured using a motion capture system and then synthesized using the computer graphics model. In the second case the motion is

generated by animating the computer graphics model. In both cases, the synthetic data provides sufficient training data to obtain good recognition results and at the same time decouples the training data from the test data far better than other approaches. In the case of the fully synthetic training data for gait type classification the acquisition process is also significantly more flexible and much faster compared to acquisition of real video for training data.

Gait continuum descriptor Walking, jogging, and running are some of the actions most often performed by people in video of real-life scenes. These actions are also a part of many action recognition systems even though running and jogging are inherently ambiguous. To deal with this ambiguity a gait continuum based on a well-established physical property of gait is introduced in chapter 5. This duty-factor representation describes the whole range of gait types with a single parameter and is used to extract information independent of the partially subjective notion of jogging and running.

Occlusion handling for interacting people Occlusion is a very difficult and important problem to address in human pose estimation, especially when people interact in the scene. Chapters 6 and 7 present two different approaches for handling occlusions in human pose estimation, both based on the pictorial structures framework. In chapter 6 a single-view approach is presented that extracts a mask of possibly occluded regions from a foreground mask. Shape based body part detection is performed within this mask and occluded body parts are found by applying strong kinematic constraints between visible and occluded body parts. The precision of the pose estimation is furthermore improved by applying a dynamic weighting between appearance information and edge information in the detection of body parts. Chapter 7 presents a multi-view approach that integrates body part probabilities from multiple cameras allowing the method to estimate the pose of a person under heavy occlusion.

1.4 Data sets for action recognition

The development of new methods within video-based analysis of human motion usually requires large amounts of video data and especially for the purpose of testing and validation there is a great need for large data sets with ground truth data. With public data sets of this kind it is also possible to directly compare new methods with state of the art. Within action recognition especially two data sets have been widely used, namely the KTH data set [24] and the Weizmann data set [3]. This has allowed for direct comparisons between many methods but with recent publications reporting recognition rates of 100% on the Weizmann data set new and more challenging data sets are needed.

Within the last five years a large amount of data sets have been produced and made publicly available all targeting different aspects of human motion analysis. A trend in these data sets is more complex actions and activities sometimes involving multiple people and also multiple views of the scene.

To give an overview of such data sets a comparative listing is presented in appendix A. This listing contains the central specifications like video resolution, number of cameras, number of subjects, actions performed, etc. A brief description of the content of the video and possible simplifications are also provided. There are other collections of data sets available but the presentation of data sets in appendix A has a strong focus on action recognition and presents a precise and consistent listing of data set characteristics. Appendix A also contains references to some relevant data set listings already available.

Another interesting feature with some of these new data sets is the release of implementations of baseline methods accompanying the data sets. This allows for a more thorough comparison and also significantly reduces the effort needed to compare already published methods to new data sets.

1.5 Publications of the thesis

The publications that result from the work of this Ph.D. thesis are listed below. The publications listed in bold appears directly in this thesis as chapters 2 to 7.

Journal papers

1. **M.B. Holte, T.B. Moeslund and P. Fihl. View-Invariant Gesture Recognition using 3D Optical Flow and Harmonic Motion Context. In *Computer Vision and Image Understanding*, December 2010**

2. P. Fihl and T.B. Moeslund. Invariant Gait Continuum Based on the Duty-Factor. In *Signal, Image and Video Processing. Springer, London*, November 2008

3. M.B. Holte, T.B. Moeslund and P. Fihl. View invariant gesture recognition using the CSEM SwissRanger SR-2 camera. In *International Journal of Intelligent Systems Technologies and Applications. Inderscience Publishers*, Great Britain, 2008

Book chapters and papers in Lecture Notes

4. **P. Fihl and T.B. Moeslund. Recognizing Human Gait Types. In *Robot Vision, ISBN 978953-3070773, In-Tech*, March 2010**

5. **P. Fihl, M.B. Holte, T.B. Moeslund. Motion Primitives and Probabilistic Edit Distance for Action Recognition. In *Gesture-Based Human-Computer Interaction and Simulation, Lecture Notes in Computer Science, vol. 5085, Springer Berlin/Heidelberg*, January 2009**

Peer reviewed conference papers

6. **P. Fihl and T.B. Moeslund. Pose Estimation of Interacting People using Pictorial Structures. In *IEEE International Conference on Advanced Video and Signal-Based Surveillance, Boston, MA, USA*, August 2010**

7. M.B. Holte, T.B. Moeslund and P. Fihl. Fusion of Range and Intensity Information for View Invariant Gesture Recognition. In *IEEE Workshop on Time of Flight Camera based Computer Vision (CVPR-workshop), Anchorage, AK, USA*, June 2008

8. P. Fihl and T.B. Moeslund. Invariant Classification of Gait Types. In *CRV 2008: Fifth Canadian Conference on Computer and Robot Vision, Windsor, ON, Canada*, May 2008

9. P. Fihl and T.B. Moeslund. Classification of Gait Types Based on the Duty-factor. In *AVSS 2007: IEEE International Conference on Advanced Video and Signal based Surveillance, London, UK*, September 2007.

10. M.B. Holte, T.B. Moeslund and P. Fihl. View Invariant Gesture Recognition using the CSEM SwissRanger SR-2 Camera. In *Dynamic 3D Imaging workshop, Heidelberg, Germany*, September 2007.

11. P. Fihl, M.B. Holte and T.B. Moeslund. Motion Primitives for Action Recognition. In *The 7th International Workshop on Gesture in Human-Computer Interaction and Simulation, Lisbon, Portugal*, May 2007.

12. **Preben Fihl, Rasmus Corlin, Sangho Park, Thomas B. Moeslund, and Mohan M. Trivedi. Tracking of Individuals in Very Long Video Sequences. *Int. Symposium on Visual Computing, Lake Tahoe, NV, USA*. In *Advances in Visual Computing, Lecture Notes in Computer Science, Vol. 4291, Springer-Verlag Berlin/Heidelberg*, 2006.**

13. Preben Fihl, Michael Boelstoft Holte, Thomas B. Moeslund, and Lars Reng. Action Recognition Using Motion Primitives and Probabilistic Edit Distance. *Andratx, Mallorca, Spain.* In *Articulated Motion and Deformable Objects, Springer-Verlag Berlin/Heidelberg*, 2006.

Non-reviewed conference papers

14. T.B. Moeslund, P. Fihl, and M.B. Holte. Action Recognition using Motion Primitives. In *The 15th Danish conference on pattern recognition and image analysis*, Copenhagen, August 2006.

Technical reports

15. **P. Fihl and S. Cosar. Full Body Pose Estimation During Occlusion using Multiple Cameras. In *Technical Report CVMT-10-02, Laboratory of Computer Vision and Media Technology, Aalborg University, Denmark*, 2010.**

16. Preben Fihl, Michael Boelstoft Holte, Thomas B. Moeslund, and Lars Reng. Action Recognition in Semi-synthetic Images using Motion Primitives. In *Technical Report CVMT-06-01, Laboratory of Computer Vision and Media Technology, Aalborg University, Denmark*, 2006.

References

[1] J.K. Aggarwal and Sangho Park. Human Motion: Modeling and Recognition of Actions and Interactions. In *3D Data Processing, Visualization and Transmission. International Symposium on*, Sept. 2004.

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial Structures Revisited: People Detection and Articulated Pose Estimation. In *Computer Vision and Pattern Recognition*, 2009.

[3] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as Space-Time Shapes. In *International Conference on Computer Vision*, Washington, DC, USA, 2005.

[4] Daniel Chen, Pi-Chi Chou, Clinton B. Fookes, and Sridha Sridharan. Multi-view Human Pose Estimation using Modified Five-point Skeleton Model. In *Int. Conference on Signal Processing and Communication Systems*, Dec. 2007.

[5] Stefano Corazza, Lars Mündermann, Emiliano Gambaretto, Giancarlo Ferrigno, and Thomas Andriacchi. Markerless Motion Capture through Visual Hull, Articulated ICP and Subject Specific Model Generation. *International Journal of Computer Vision*, 87, 2010.

[6] R. Ronfard D. Weinland and E. Boyer. Free Viewpoint Action Recognition using Motion History Volumes. In *Computer Vision and Image Understanding*, 104(2):249-257, 2006.

[7] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *Computer Vision and Pattern Recognition*, San Diego, CA, USA, June 2005.

[8] M. Enzweiler and D.M. Gavrila. A Mixed Generative-discriminative Framework for Pedestrian Classification. In *Computer Vision and Pattern Recognition*, June 2008.

[9] M. Enzweiler and D.M. Gavrila. Monocular Pedestrian Detection: Survey and Experiments. *Pattern Analysis and Machine Intelligence*, 31(12):2179 –2195, Dec. 2009.

[10] Vittorio Ferrari, Manuel J. Marn-Jimnez, and Andrew Zisserman. Progressive search space reduction for human pose estimation. In *Computer Vision and Pattern Recognition*, 2008.

[11] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua. Multi-camera People Tracking with a Probabilistic Occupancy Map. *Transactions on Pattern Analysis and Machine Intelligence*, 30(2), February 2008.

[12] Weiming Hu, Tieniu Tan, Liang Wang, and S. Maybank. A Survey on Visual Surveillance of Object Motion and Behaviors. *Systems, Man, and Cybernetics, Transactions on*, 34(3), Aug. 2004.

[13] Xiaofei Ji, Honghai Liu, Yibo Li, and David Brown. Visual-Based View-Invariant Human Motion Analysis: A Review. In *Knowledge-Based Intelligent Information and Engineering Systems*, volume 5177 of *LNCS*. Springer Berlin / Heidelberg, 2008.

[14] H. Jiang. Human Pose Estimation Using Consistent Max-Covering. In *International Conference on Computer Vision*, 2009.

[15] B. Leibe, E. Seemann, and B. Schiele. Pedestrian Detection in Crowded Scenes. In *Computer Vision and Pattern Recognition*, San Diego, CA, USA, June 2005.

[16] Haibin Ling and David W. Jacobs. Shape Classification Using the Inner-Distance. *Transactions on Pattern Analysis and Machine Intelligence*, 29, 2007.

[17] Yusuke Matsumoto, Toshikazu Wada, Shuichi Nishio, Takehiro Miyashita, and Norihiro Hagita. Scalable and Robust Multi-people Head Tracking by Combining Distributed Multiple Sensors. *Intelligent Service Robotics*, 3(1), 2010.

[18] J. R. Mitchelson and A. Hilton. Simultaneous Pose Estimation of Multiple People using Multiple-View Cues with Hierarchical Sampling. In *British Machine Vision Conference*, 2003.

[19] T.B. Moeslund, A. Hilton, and V. Krüger. A Survey of Advances in Vision-Based Human Motion Capture and Analysis. *Journal of Computer Vision and Image Understanding*, 104(2-3), 2006.

[20] Ofir Pele and Michael Werman. The Quadratic-Chi Histogram Distance Family. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision - ECCV 2010*, volume 6312 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2010.

[21] Ronald Poppe. Vision-based Human Motion Analysis: An Overview. *Computer Vision and Image Understanding*, 108(1-2), 2007. Special Issue on Vision for Human-Computer Interaction.

[22] Ronald Poppe. A Survey on Vision-based Human Action Recognition. *Image and Vision Computing*, 28(6), 2010.

[23] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, 40, November 2000.

[24] C. Schüldt, I. Laptev, and B. Caputo. Recognizing Human Actions: a Local SVM Approach. In *International Conference on Pattern Recognition*, 2004.

[25] Leonid Sigal and Michael J. Black. Measure Locally, Reason Globally: Occlusion-sensitive Articulated Pose Estimation. In *Computer Vision and Pattern Recognition*, 2006.

[26] R. Souvenir and J. Babbs. Learning the Viewpoint Manifold for Action Recognition. In *Computer Vision and Pattern Recognition*, Anchorage, Alaska, June 2008.

Chapter 2

Foreground Segmentation

This chapter consists of the paper "Tracking of Individuals in Very Long Video Sequences" [A]. The paper presents a robust background subtraction method which is utilized in a appearance based tracker. The background subtraction and the tracking is evaluated on a ten hour long video showing stable segmentation and tracking results over the whole duration of the video. This background subtraction algorithm is also used as a pre processing step for action recognition and pose estimation in chapters 5 and 6.

References

A. P. Fihl, R. Corlin, S. Park, T.B. Moeslund, and M.M. Trivedi. Tracking of Individuals in Very Long Video Sequences. In *International Symposium on Visual Computing, Advances in Visual Computing, LNCS, Vol. 4291, Springer-Verlag Berlin/Heidelberg*, Lake Tahoe, Nevada, USA, November 6-8 2006.

Tracking of Individuals in Very Long Video Sequences

P. Fihl, R. Corlin, S. Park, T.B. Moeslund, and M.M. Trivedi

Abstract

In this paper we present an approach for automatically detecting and tracking humans in very long video sequences. The detection is based on background subtraction using a multi-mode Codeword method. We enhance this method both in terms of representation and in terms of automatically updating the background allowing for handling gradual and rapid changes. Tracking is conducted by building appearance-based models and matching these over time. Tests show promising detection and tracking results in a ten hour video sequence.

2.1 Introduction

Visual analysis of humans has a number of applications ranging from automatic surveillance systems to extracting pose parameters for realistically character animation in movies. Automatic surveillance systems observe humans at a distance and in various environments. Furthermore, these systems should, as opposed to e.g., motion capture systems, work completely autonomous and for long periods of time.

The foundation of many surveillance systems is a good detection and tracking of humans in a video sequence. These issues have received much attention in the last decade or so [5, 11, 20, 13]. The detection problem (aka the figure-ground segmentation problem) is typically done using shape, motion, depth, background detection, or appearance [10, 17, 15, 6, 16, 14]. When the scene of interest contains individuals that are allowed to occlude each other, the tracking of individuals is inherently difficult and using an appearance-based model for each individual is often the preferred approach.

In this work we consider situations where occlusion can occur and we therefore follow the appearance-based approach. Our aim is continuous detection and tracking over very long periods of time as opposed to other approaches mostly evaluated on short video sequences. Concretely, we first develop and use an advanced background subtraction algorithm in order to handle the figure-ground segmentation problem. The result is a silhouette of each individual in the scene - section 2.2. Next we use an appearance-based model to represent each individual. A good model is obtained by using some of the results from research on modeling interacting people. We then present a scheme for matching appearance-based models over time - section 2.3. In section 2.4 we present tracking results of several hours of continuous video and in section 2.5 a conclusion is given.

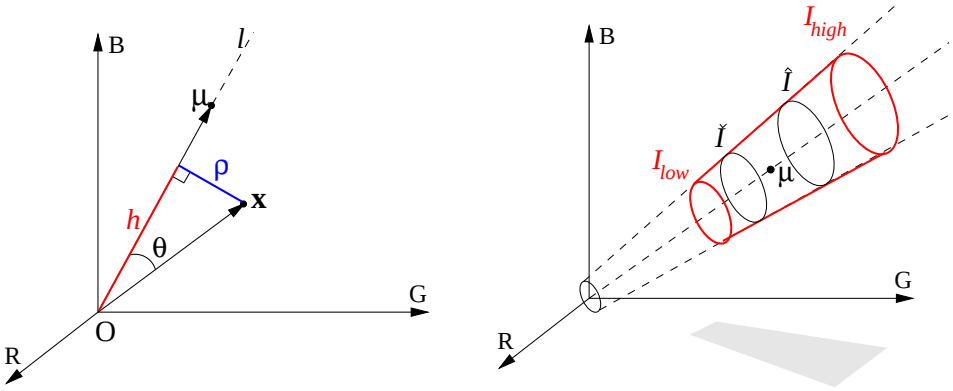
2.2 Figure-ground segmentation

The first step in our tracking algorithm is to separate the foreground (humans) from the background, i.e., the figure-ground segmentation problem. We do this using a background subtracting approach inspired by [8].

2.2.1 Background representation

We apply the Codeword approach of [8], which has shown to perform better than Gaussian mixture models [16] and other well-known methods [3, 5] in terms of both speed and sensitivity [2].

The representation of a background pixel in the codeword approach [8] is based on the representation from [7]. Here color and intensity are represented independently and a background pixel is represented as a vector in the RGB-cube, $\boldsymbol{\mu}$. The distance, in terms of color, ρ, from a new pixel, $\boldsymbol{x}$, to the background model is measured as the



(a) The color and intensity representation.

(b) A codeword representation.

Figure 2.1: Illustration of the representations used in the background subtraction.

perpendicular distance to the vector. The difference in intensity is measured along the vector and denoted, h, see figure 2.1. In the work by [8] a cylinder centered around the vector represents a codeword for this particular pixel and all pixel values inside the cylinder are classified as background. During a training phase a number of codewords are learned for each pixel and together these are denoted the codebook for this particular pixel. This is a fast and robust approach due to the multi-mode nature of the representation. However, since all color vectors go through the origin of the color-cube a more correct representation is to form a truncated cone around each learned background vector. In this way the different colors inside the codeword actually corresponds to the same colors with different intensity. In figure 2.1 our representation of a codeword is shown, where $\check{I}$ and $\hat{I}$ are the minimum and maximum values found during training, and I_{low} and I_{high} are where the cone is truncated in order to allow additional values, e.g., due to shadows [1].

2.2.2 Background initialization

A key issue in successful background subtraction is to learn a good model of the background during an initialization phase. If no moving objects are present in the scene this is obviously easier. But a more general approach is to allow for moving objects. If a pixel is covered by moving objects in less than 50% of the learning period then a median filter can be applied [5]. A different method is first to divide the training sequence into temporal subintervals with similar values - assumed to belong to the background and then find the subinterval with the minimum average motion and only use these pixels for model initialization [4, 18]. In this work we follow the approach by [8], which also works along these lines of reasoning.

During the initialization phase each new pixel is either assigned to an already existing codeword (which is updated accordingly) or a new codeword is created. This will produce codewords for non-background pixels, but these are handled by temporal filtering using the so-called *maximum negative run-length* (MNRL). This measures the longest time interval where no pixel values have been assigned to the codeword. After the training period all codewords with a too large maximum negative run-length will be removed from the background model, i.e., this process allows for moving objects during the training period, see [8] for details.

2.2.3 Background updating

Using multiple codewords for each pixel allows modeling of very dynamic scenes, but only the variation that is present in the training period will be modeled by the codebook background method as described in [8][1]. For the background subtraction to work for several hours it is necessary also to handle the changes in the background that occurs after the initialization phase. Two different types of changes need to be handled.

- **Gradual changes** do not change the appearance of the background much from one frame to the next. The accumulated change over time can however be large, e.g., the effect of the changing position of the sun during a day.
- **Rapid changes** cause significant changes in the background from one frame to the next. Background objects that are moved or significant changes in the motion patterns of vegetation caused by gusting winds will for example cause rapid changes.

To handle the gradual changes we apply a simple continuous model:

$$\boldsymbol{\mu}_{t+1} = (1-\alpha)\cdot\boldsymbol{\mu}_t + \alpha\cdot\boldsymbol{x}_t \quad , \quad 0 \leq \alpha \leq 1 \tag{2.1}$$

where μ_t is the codeword, μ_{t+1} is the updated codeword, and x_t is the current pixel value. Based on experiences in dynamic outdoor environments α is typical 0.05 to 0.15.

Only the activated codeword in each codebook is updated with this process. Consider the situation where a pixel is sometimes occupied by a green tree branch and sometimes occupied by a red wall, e.g., due to wind. Only the codeword modeling the branch should be updated when the branch occupies the pixel, since the color of surfaces with different orientation, texture, and color change in different ways with changing illumination. The change of the codeword that models the wall cannot be found from the color change of the branch.

[1] It should be noted that at the time when this work was finished [1] the authors of [8] published a more advanced version of their codebook algorithm [9], which is somewhat similar to our background subtracting approach.

Pixels falsely classified as background will lead to codewords that are updated to model foreground instead of background. Therefore only pixels describing stable background will be updated. Stable background is defined as a pixel that has been classified as background for the last j frames, where j typically is 10-15. As for α this interval is based on experiences in dynamic outdoor environments. The performance of the background updating is not very sensitive to neither j nor α within these intervals.

Equation 2.1 cannot handle rapid changes and therefore new codewords are learned during run-time. For example, in a situation when a car is parked in the scene it is treated (correctly) as a foreground object. However, while the car is parked we want it to become part of the background so we can detect new foreground objects that move in front of the car. We do this in the following way. Each time a pixel is classified as foreground we create a new codeword, denoted a *training codeword*. If this codeword has a small MNRL within the next n frames we conclude that this codeword does indeed represent a new background and we make it a *temporary codeword*[2]. If the MNRL is big we delete this training codeword. Temporary codewords that become inactive (measured by their MNRL) are deleted, e.g., if the parked car starts to move again. So in each frame a pixel value is matched against the codewords from the "real" background (learned during initialization), the temporary codewords and the training codewords, in that order. If a match is found, the respective codeword is updated using equation 2.1.

2.2.4 Bounding box representation

After an image is processed by the background subtraction process we remove noise (false positives) using a median filter. Sometimes false negatives result in the silhouette being split into smaller blobs. We therefore investigate the size and proximity of the bounding boxes of each blob and try to merge them into bounding boxes each representing one human [1]. The silhouettes in the merged bounding boxes are compared to a simple body model to distinguish humans from small blobs of noise. The silhouette of a person can roughly be described by an ellipse, and our body model defines limits for the ratio between the major and minor axes of the ellipse, the slope of the major axis, the fidelity between the ellipse and silhouette, and the area of the silhouette. A silhouette complying with all limits of the body model will be accepted as a person otherwise it is considered as noise. To avoid the problem with a person producing multiple enter/exit results when the area of the silhouette is close to the limit of the body model we utilize a hysteresis threshold [1]. To get a correct initialization of our appearance model we need to ensure that a person is completely inside the field-of-view before we accept the new person and we therefore introduce an entry zone around the image border [1]. To summaries, after the above processes we are left with a number of bounding boxes each containing the silhouette of one person.

[2]Pixels that belong to a training codeword are classified as foreground whereas pixels belonging to a temporary codeword are classified as background.

2.3 Tracking

2.3.1 Representation

We model the appearance of a person by dividing it into two regions which are modeled separately: the upper body (not including the head) and the lower body [13]. Due to the nature of the method the regions are not simply found as a ratio of the bounding box as seen in, e.g., [12, 14, 19] but are found by dividing the body into a set of blobs that are similar in color and spatially connected, and then grouping these blobs into an upper body and a lower body using a ratio of the bounding box as a guideline.

The blobs are initialized by labeling pixels with similar color in the foreground to the same class. To do this the Expectation Maximization (EM) algorithm is used to first learn the gaussian distributions of color classes in the foreground followed by a classification of the pixels to these classes. The labeling of pixels to color classes is carried out by a Maximum Likelihood estimation.

When the pixels have been classified in this way the classes are not necessarily spatially connected, e.g. the dark hair of a person could be assigned to the same class as a dark pair of shoes or simply a checkered shirt could consists of many spatially disconnected classes of the same color. To make sure that each blob represents a region of connected pixels a relabeling is done by making a connected component analysis. This is done by finding the contours of all disconnected regions for the pixels in each color class separately and giving all pixels within the boundary of these contours a unique label.

To avoid over-segmentation of the foreground similar blobs are merged. Blob similarity is evaluated using four criteria [1] and two blobs are merged if either the first criterion is true or if the three remaining criteria are all true: 1) a blob is completely surrounded by another blob, 2) two blobs are adjacent, 3) two blobs share a large border, 4) two blobs are similar in terms of color. By use of these criteria the number of blobs is reduced considerably and a set of blobs that are expected to represent relatively stable parts of the foreground is obtained.

To define the merged blobs as either upper body or lower body we use ratios of the bounding box as a guideline. The bounding box is divided into three regions as shown in figure 2.2 [1]. All blobs with centroid in the range from 0 to 0.55 times the height of the bounding box will define the lower body and blobs with centroid from 0.55 to 0.84 times the height of the bounding box will define the upper body. This way the border between upper and lower body will not be a straight line but follow the borders of dissimilar blobs. The final features representing a person (or silhouette) are listed in the feature vector $\vec{m}$:

$$\vec{m} = [\mu_x, \mu_y, \mu_{H\_upper}, \mu_{S\_upper}, \mu_{H\_lower}, \mu_{S\_lower}]^T \tag{2.2}$$

where μ_x and μ_y are the mean position or center of mass of the given person. The last

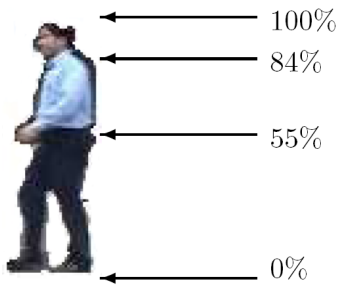


Figure 2.2: Ranges for partitioning the silhouette of a person into head, upper body and lower body in relation to the height of the person. Note that the hand is assigned to the upper body even though its located below the 0.55 line. This is due to the complex merging process described above.

four parameters represent the mean color of the upper and lower body respectively in terms of hue and saturation [1].

2.3.2 Matching

The matching of identities is performed by calculating the dissimilarity in terms of the Mahalanobis distance[3] between all extracted silhouettes in the current frame, indexed by i, and all known identities that have been tracked to this frame, indexed by j:

$$\Delta_{ij} = (\vec{m}_i - \vec{m}_j)^T(\Pi_i + \Pi_j)^{-1}(\vec{m}_i - \vec{m}_j) \tag{2.3}$$

where Π_i represents the between class covariance of all body models in the current frame and Π_j the between class covariance for the identities that the body models are being matched to. These are pooled in order to compensate for the differences in the variations of the body models and the identities they are being matched to. To simplify the calculations only the diagonal of these covariances have been used. The between class variances have been calculated as follows:

$$\Pi = \frac{1}{k}\sum_k (\vec{\mu}_k - \vec{\mu}_{all})(\vec{\mu}_k - \vec{\mu}_{all})^T \tag{2.4}$$

where $\vec{\mu}_k$ is the mean value of the k^{th} body model/identity and $\vec{\mu}_{all}$ is the mean of all body models/identities present in the given region [1].

[3]Note that a weight is assigned to each feature in order to balance the positional features and the appearance features. I.e., 0.3 for the positional features and 0.1 for the appearance features. The weights are omitted from the equation for clarity.

2.4 Results

The presented system has been tested at two levels. The figure-ground segmentation have been tested to show its performance on very long video sequences (10 hours). The bounding box representation and the tracking have been tested on the output of the figure-ground segmentation to show the system's overall capability to track people.

2.4.1 Test of figure-ground segmentation

The video used for the test is a 10 hour video with a frame rate of 30 frames per second. The video is captured from 9.15 AM to 7.15 PM. and the scene contains several challenging situations in the context of figure-ground segmentation, i.e. illumination changes, non-static background, shadows, puddles, and foreground camouflage (see figure 2.4).

To calculate the false rejection rate (percentage of foreground pixels falsely classified as background) and the false acceptance rate (percentage of background pixels falsely classified as foreground) a set of 93 frames containing foreground objects (people) have been sampled from the whole time span. The images used are the binary foreground mask obtained from the figure-ground segmentation filtered with the median filter. For each of the sampled frames the foreground region was marked by hand (based on the original input frame) and used as the ground truth.

The calculation of the false acceptance rate is based on the above mentioned frames in addition to a set of frames containing only background. The frames containing only background were added since only a limited number of frames actually contain people, and the additional frames would give a more representative result for the whole video. The frames containing only background were sampled every 1000 frames. When sampling every 1000 frames some of the frames contained people, but these frames were discarded from the set giving a total of 971 frames.

Figure 2.3(a) shows FRR (false rejection rate) and FAR (false acceptance rate) in percent as a function of time. FRR is in the range [0%;62.4%] with mean value 8.45% and standard deviation 13.43. The black line represents the best linear fit of the FRR samples in the least-squares sense and shows a slightly increasing tendency in FRR. The increase does however not mean that the performance of the background subtraction decreases over time. The increase in FRR is caused by the low overall illumination level at the end of the day which causes the problem of foreground camouflage to increase. The FAR is in the range [0%;4.9%] with mean value 0.14% and standard deviation 0.33. The FAR shows a slightly decreasing tendency over time. The mean FAR of 0.14% shows that the background subtraction in general effectively models the background and adapts to the changes present in the test video. The performance of the background subtraction in terms of FAR in not dependent on how many hours it has been running, but on the type of changes that happens in the scene, and the background subtraction automatically recovers from changes that are not directly handled by the model.

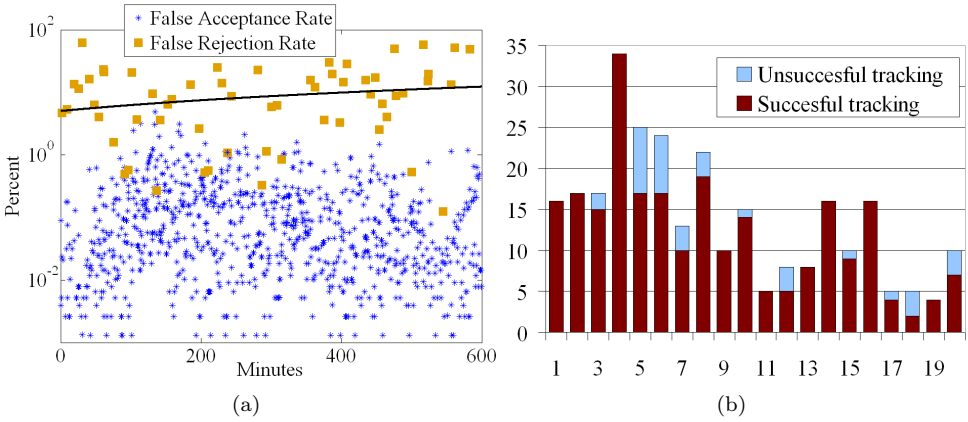


Figure 2.3: Results from the 10 hour test video. (a): The FAR and FRR in percent as a function of time. The black line is the best linear fit of the FRR samples in the least-squares sense. Note that the y-axis is logarithmic. (b): The number of successful and unsuccessful tracks. Each column covers 30 minutes.

2.4.2 Test of tracking

The 10 hours of test video contains 267 persons that move through the scene[4]. The tracking result of each person has been evaluated to see if the system successfully identifies each person and tracks the person.

The system identifies and tracks 247 persons successfully. 20 persons are not tracked correctly and the system further identifies 15 blobs of noise as persons which gives a total of 35 errors. The overall successful tracking rate yields 86.9%. Figure 2.3(b) shows the tracking result for each 30 minutes interval.

Figure 2.3(b) indicates that the performance of the tracking is independent of the number of hours the system has been running. The number of tracking errors is most remarkable in the 3rd and 9th hour. This is due to rapid background variations and low overall illumination, respectively, and not because the system has been running for several hours. Figure 2.4 shows examples of tracking results (both successful and unsuccessful) when multiple people are in the scene.

The errors that occur during tracking can be explained by either *noisy foreground segmentation* (24 errors) or *insufficient tracking or appearance model* (11 errors). The errors originating from noisy foreground segmentation are mainly due to moving vegetation and strong shadows that gets identified as humans. The errors originating from insufficient tracking or appearance model often occur when two persons move

[4]People that never enter the field-of-view completely or people that move through the scene in groups are not included in this number or the test.

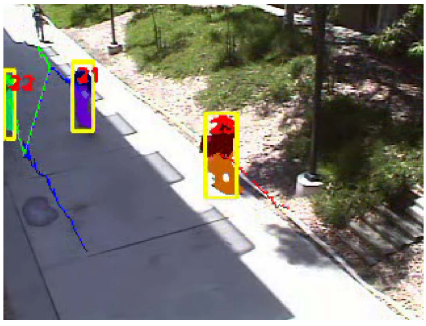
(a) 9.25 AM. The box shows an example of a region with foreground camouflage. Notice the puddles on the ground.

(b) The two persons are tracked correctly even though they move near each other.

(c) 3.05 PM. The boxes show examples of regions with strong shadows. Furthermore, other shadows move rapidly because of the wind.

(d) The three people are tracked correctly until the two persons to the left get too close to each other.

(e) 7.05 PM. The overall illumination level changes significantly from morning to the afternoon and again from the afternoon to the evening



(f) The tracks of the two persons to the left are swapped.

Figure 2.4: Left: Examples of the changes in the scene during the 10 hours test video. Right: Tracking results.

close past each other (resulting in switching identities) or when strong shadows makes the silhouette of a person non-elliptic. The effect of both types of errors can possibly be reduced by including temporal information.

2.5 Conclusion

In this paper we have presented a system to do figure-ground segmentation and tracking of people in an outdoor scene continuously for several hours.

The system has been thoroughly tested on 10 hours of continuous video containing multiple difficult situations. The system was able to automatically update the background model allowing for tracking people with a success rate of 86.9%, and we believe that this number can be increased with relatively simple improvements to the tracking algorithm. To our knowledge this is the first system to present results on continuous tracking in very long video sequences.

References

[1] P.F. Andersen and R. Corlin. Tracking of Interacting People and Their Body Parts for Outdoor Surveillance. Master's thesis, Laboratory of Computer Vision and Media Technology, Aalborg University, Denmark, 2005.

[2] T.H. Chalidabhongse, K. Kim, D. Harwood, and L. Davis. A Perturbation Method for Evaluating Background Subtraction Algorithms. In *Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Beijing, China, Oct 15-16 2005.

[3] A. Elgammal, D. Harwood, and L. Davis. Non-Parametric Model for Background Subtraction. In *European Conference on Computer Vision*, Dublin, Ireland, June 2000.

[4] D. Gutchess, M. Trajkovic, E.C. Solal, D. Lyons, and A. Jain. A Background Model Initialization Algorithm for Video Surveillance. In *International Conference on Computer Vision*, Vancouver, Canada, July 9-12 2001.

[5] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: Real-Time Surveillance of People and Their Activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 2000.

[6] K. Hayashi, M. Hashimoto, K. Sumi, and K. Sasakawa. Multiple-Person Tracker with a Fixed Slanting Stereo Camera. In *International Conference on Automatic Face and Gesture Recognition*, Seoul, Korea, May 17-19 2004.

[7] T. Horprasert, D. Harwood, and L.S. Davis. A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection. In *IEEE ICCV'99 FRAME-RATE WORKSHOP*, Corfu, Greece, September 1999.

[8] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis. Background modeling and subtraction by codebook construction. In *IEEE International Conference on Image Processing (ICIP)*, 2004.

[9] K. Kim, T.H. Chalidabhongse, D. Harwood, and L. Davis. Real-Time Foreground-Background Segmentation using Codebook Model. *Real-Time Imaging*, 11(3), 2005.

[10] B. Leibe, E. Seemann, and B. Schiele. Pedestrian Detection in Crowded Scenes. In *Computer Vision and Pattern Recognition*, San Diego, CA, USA, June 20-25 2005.

[11] S.J. McKenna, S. Jabri, Z. Duric, and H. Wechsler. Tracking Interacting People. In *The fourth International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, March 2000.

[12] A. Mittal and L.S. Davis. M2Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene. *International Journal of Computer Vision*, 51(3):189–203, 2003.

[13] S. Park and J.K. Aggarwal. Simultaneous tracking of multiple body parts of interacting persons. *Computer Vision and Image Understanding*, 102(1), 2006.

[14] D. Roth, P. Doubek, and L.V. Gool. Bayesian Pixel Classification for Human Tracking. In *IEEE Workshop on Motion and Video Computing (MOTION'05)*, Breckenridge, Colorado, Jan 2005.

[15] H. Sidenbladh. Detecting Human Motion with Support Vector Machines. In *International Conference on Pattern Recognition*, Cambridge, UK, Aug 2004.

[16] C. Stauffer and W.E.L. Grimson. Adaptive Background Mixture Models for Real-Time Tracking. In *Computer Vision and Pattern Recognition*, Santa Barbara, CA, USA, June 1998.

[17] P. Viola, M.J. Jones, and D. Snow. Detecting Pedestrians Using Patterns of Motion and Appearance. *International Journal of Computer Vision*, 63(2), 2005.

[18] H. Wang and D. Suter. Background Initialization with a New Robust Statistical Approach. In *Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Beijing, China, Oct 15-16 2005.

[19] C. Yang, R. Duraiswami, and L. Davis. Fast Multiple Object Tracking via a Hierarchical Particle Filter. In *International Conference on Computer Vision*, Beijing, China, Oct 15-21 2005.

[20] T. Zhao and R. Nevatia. Tracking Multiple Humans in Crowded Environments. In *Computer Vision and Pattern Recognition*, Washington DC, USA, June 2004.

Chapter 3

2D Human Gesture Recognition

This chapter consists of the paper "Motion Primitives and Probabilistic Edit Distance for Action Recognition" [A]. The paper presents a key-frame based method to recognize arm gestures. The gestures represent typical signaling between people over long distances. The method uses traning data which is synthezised with a computer graphics model of a human and achieve good recognition results on real test video.

References

A. P. Fihl, M.B. Holte, T.B. Moeslund. Motion Primitives and Probabilistic Edit Distance for Action Recognition. In *Gesture-Based Human-Computer Interaction and Simulation, Lecture Notes in Computer Science, vol. 5085, Springer Berlin/Heidelberg,* January 2009

Motion Primitives and Probabilistic Edit Distance for Action Recognition

P. Fihl, M.B. Holte and T.B. Moeslund

Abstract

The number of potential applications has made automatic recognition of human actions a very active research area. Different approaches have been followed based on trajectories through some state space. In this paper we also model an action as a trajectory through a state space, but we represent the actions as a sequence of temporal isolated instances, denoted primitives. These primitives are each defined by four features extracted from motion images. The primitives are recognized in each frame based on a trained classifier resulting in a sequence of primitives. From this sequence we recognize different temporal actions using a probabilistic Edit Distance method. The method is tested on different actions with and without noise and the results show recognition rates of 88.7% and 85.5%, respectively.

3.1 Introduction

Automatic recognition of human actions is a very active research area due to its numerous applications. As opposed to earlier the current trend is not as much on first reconstructing the human and the pose of his/her limbs and *then* do the recognition on the joint angle data, but rather to do the recognition directly on the image data, e.g., silhouette data [20, 21, 23] or spatio-temporal features [1, 4, 15].

Common for these approaches is that they represent an action by image data from all frames constituting the action, e.g., by a trajectory through some state-space or a spatio-temporal volume. This means that the methods in general require that the applied image information can be extracted reliably in every single frame. In some situations this will not be possible and therefore a different type of approach has been suggested. Here an action is divided into a number of smaller temporal sequences, for example movemes [6], atomic movements [7], states [5], dynamic instants [16], examplars [11], behaviour units [9], and key-frames [8]. The general idea is that approaches based on finding smaller units will be less sensitive compared to approaches based on an entire sequence of information.

For some approaches the union of the units represents the entire temporal sequence, whereas for other approaches the units represent only a subset of the original sequence. In Rao *et al.* [16] dynamic hand gestures are recognized by searching a trajectory in 3D space (x and y-position of the hand, and time) for certain dynamic instants. Gonzalez *et al.* [8] look for key-frames for recognizing actions, like walking and running. Approaches where the entire trajectory (one action) is represented by a number of subsequences are Barbic *et al.* [2] for full body motion, where probabilistic PCA is used for finding transitions between different behaviors, and Bettinger *et al.* [3] where likelihoods are used to separate a trajectory into sub-trajectories. These sub-trajectories are modeled by Gaussian distributions each corresponding to a temporal primitive.

3.2 Paper content and system design

In this paper we address action recognition using temporal instances (denoted primitives) that only represent a subset of the original sequence. That is, our aim is to recognize an action by recognizing only a few primitives as opposed to recognition based on the entire sequence (possibly divided into sub-trajectories).

Our approach is based on the fact that an action will always be associated with a movement, which will manifest itself as temporal changes in the image. So by measuring the temporal changes in the image the action can be inferred. We define primitives as temporal instances with a significant change and an action is defined as a set of primitives. This approach allows for handling partly corrupted input sequences and, as we shall see, does not require the lengths, the start point, nor the end point to be known, which is the case in many other systems.

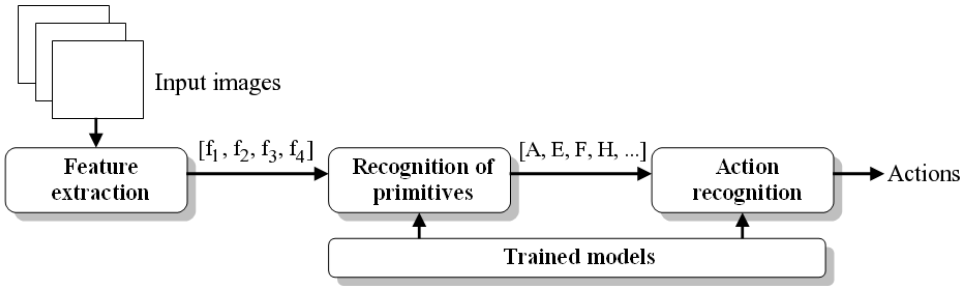


Figure 3.1: System overview.

Measuring the temporal changes can be done in a number of ways. We aim at primitives that are as independent on the environment as possible. Therefore, we do not rely on figure-ground segmentation using methods like background subtraction or personalized models etc. Instead we define our primitives based on image subtraction. Image subtraction has the benefit that it measures the change in the image over time and can handle very large changes in the environment.

Concretely we represent our primitives by four features extracted from a motion-image (found by image subtraction). In each frame the primitive, if any, that best explains the observed data is identified. This leads to a discrete recognition problem since a video sequence will be converted into a string containing a sequence of symbols, each representing a primitive. After pruning the string a probabilistic Edit Distance classifier is applied to identify which action best describes the pruned string. The system is illustrated in figure 3.1.

The actions that we focus on in this work are five one-arm gestures, but the approach can with some modifications be generalized to body actions. The actions are inspired by [10] and can be seen in figure 3.2.

The paper is structured as follows. In section 3.3 we describe how our features are extracted. In section 3.4 we describe how we recognize the primitives, and in section 3.5 we describe how we recognize the actions. In section 3.6 the approach is evaluated on a number of actions and in section 3.7 the approach is discussed.

3.3 Feature extraction

Even though image subtraction only provides crude information it has the benefit of being rather independent to illumination changes and clothing types and styles. Furthermore, no background model or person model is required. However, difference images suffer from "shadow effects" and we therefore apply double difference images, which are known to be more robust [22]. The idea is to use three successive images in order to create two difference images. These are thresholded and ANDed together. This ensures that only pixels that have changed in both difference images

are included in the final output. The motion extraction process is illustrated in figure 3.3.

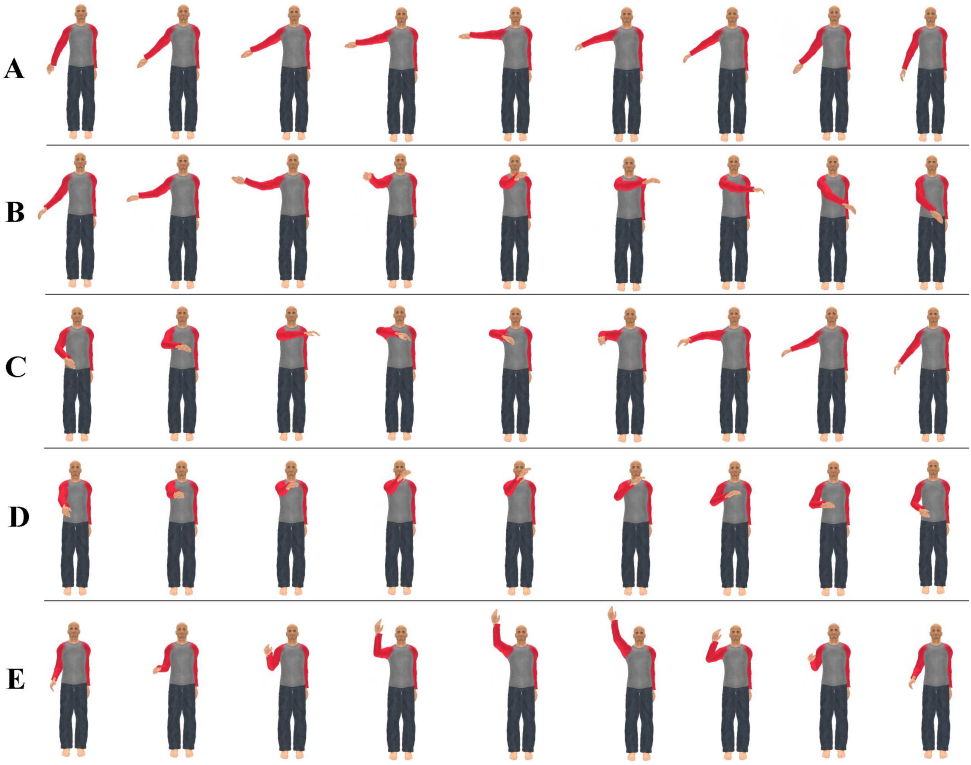


Figure 3.2: Samples from the five actions. The following describes the actions as seen from the person performing the action. **A - Point right:** A stretched arm is raised to a horizontal position pointing right, and then lowered down. **B - Move left:** A stretched arm is raised to a horizontal position pointing right. The arm is then moved in front of the body ending at the right shoulder, and then lowered down. **C - Move right:** Right hand is moved up in front of the left shoulder. The arm is then stretched while moved all the way to the right, and then lowered down. **D - Move closer:** A stretched arm is raised to a horizontal position pointing forward while the palm is pointing upwards. The hand is then drawn to the chest, and lowered down. **E - Raise arm:** The arm is moved along the side of the person, stretched above the head, and then lowered again.

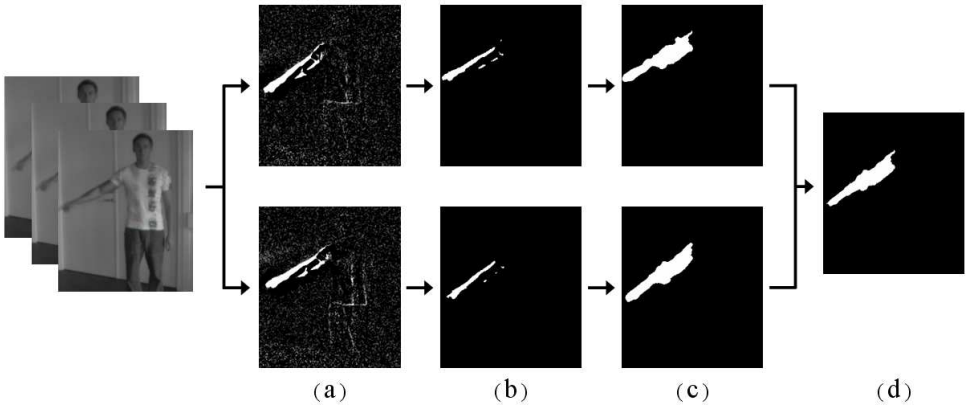


Figure 3.3: An illustration of the motion extraction process. (a) Difference images are calculated from a set of three input frames yielding noisy gray scale images. (b)+(c) The hysteresis thresholds T_1 and T_2 are applied. (d) The two thresholded images from (c) are ANDed together resulting in a single connected motion-cloud.

Multiple steps between the three successive images used to generate the double difference image have been investigated (frames 1-2-3, frames 1-3-5, and frames 1-4-7, etc.). The approach is rather invariant to this choice, i.e., invariant to the frame-rate and the execution speed of the actions. Frames 1-3-5 are used in this work.

When doing arm gestures the double difference image will roughly speaking contain a "motion-cloud". However, noise will also be present. Either from other movements, e.g., the clothes on the upper body when lifting the arm (false positives), or the motion-cloud will be split into a number of separate blobs, e.g., due to the shirt having a uniform color (false negatives). Since the two noise sources "work against each other", it is difficult to binarize the difference image. We therefore apply a hysteresis principle consisting of two thresholds T_1 and T_2 with $T_1 > T_2$. For all difference pixels above T_1 we initiate a region growing procedure which continues to grow until the pixel values falls below T_2, see figure 3.4.

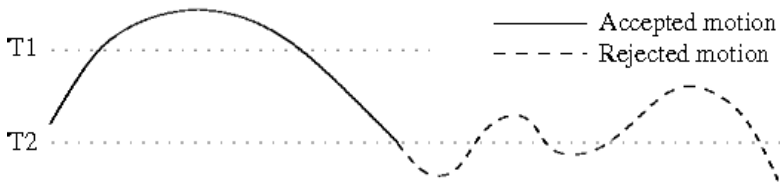


Figure 3.4: An illustration of the hysteresis with an upper threshold T_1 and a lower threshold T_2. The figure illustrates the advantage of the hysteresis, where most of the "motion-blob" of interest is accepted while the smaller "noise-blobs" are rejected.

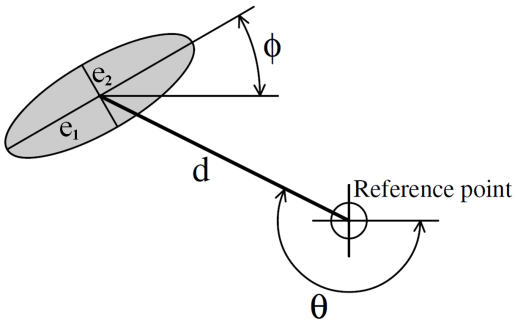


Figure 3.5: An illustration of the features used for describing the motion-cloud. See text for details.

The resulting connected motion components are further sorted with respect to their size to obtain robustness towards noise. This hysteresis threshold helps to ensure that noisy motion-clouds are not broken up into multiple fragments and at the same time eliminates small noisy motion blobs. The result is one connected motion-cloud.

We model the motion-cloud compactly by an ellipse. The length and orientation of the axes of the ellipse are calculated from the Eigen-vectors and Eigen-values of the covariance matrix defined by the motion pixels. We use four features to represent the motion cloud. They are independent of image size and the person's size and position in the image. To ensure the scale invariance two of the features are defined with respect to a reference point currently defined manually as the center of gravity of the person. The features are illustrated in figure 3.5 and defined as follows. Feature 1 is the eccentricity of the ellipse defined as the ratio between the axes of the ellipse (e_2/e_1). Feature 2 is the orientation of the ellipse defined as the angle between the x-axis of the image and the major axis of the ellipse (ϕ). Feature 3 is the size of the ellipse defined as the ratio between the length of the major axis and the distance from the reference point to the center of the ellipse (e_1/d). Feature 4 is the angle between the x-axis of the image through the reference point and the line from the center of the ellipse to the reference point (θ).

3.4 Recognition of primitives

Each incoming frame is represented by the four extracted features described above. This feature vector is then classified as a particular primitive or as noise with a Mahalanobis classifier. From a set of training examples we extract representative feature vectors for each primitive. The primitives are then formed by the mean and covariance of the representative feature vectors, see below. The four features are not equally important and therefore weighted in accordance with their importance in classification. Experiments yielded features 2 and 4 as the most descriminative

and feature 1 as the least descriminative. This gives the following classifier for recognizing a primitive at time t:

$$\text{Primitive}(t) = \arg\min_i \left[(\vec{W} \cdot (\vec{f_t} - \vec{p_i}))^T \Pi_i^{-1} (\vec{W} \cdot (\vec{f_t} - \vec{p_i})) \right] \tag{3.1}$$

where $\vec{f_t}$ is the feature vector estimated at time t, $\vec{p_i}$ is the mean vector of the ith primitive, Π_i is the covariance matrix of the ith primitive, and $\vec{W}$ contains the weights and are included as an element-wise multiplication.

The classification of a sequence can be viewed as a trajectory through the 4D feature space where, at each time-step, the closest primitive (in terms of Mahalanobis distance) is found. To reduce noise in this process we introduce a minimum Mahalanobis distance in order for a primitive to be considered in the first place. Furthermore, to reduce the flickering observed when the trajectory passes through a border region between two primitives we introduce a hysteresis threshold. It favors the primitive recognized in the preceding frame over all other primitives by modifying the individual distances. The classifier hereby obtains a "sticky" effect, which handles a large part of the flickering.

After processing a sequence the output will be a string with the same length as the sequence. An example is illustrated in equation 3.2. Each letter corresponds to a recognized primitive (see figure 3.7) and Ø corresponds to time instances where no primitives are below the minimum required Mahalanobis distance. The string is pruned by first removing 'Ø's, isolated instances, and then all repeated letters, see equation 3.3. A weight is generated to reflect the number of repeated letters (this is used below).

$$\text{String} = \{\text{Ø}, \text{Ø}, B, B, B, B, B, E, A, A, F, F, F, F, \text{Ø}, D, D, G, G, G, G, \text{Ø}\} \tag{3.2}$$

$$\text{String} = \{B, A, F, D, G\} \tag{3.3}$$

$$\text{Weights} = \{5, 2, 4, 2, 4\} \tag{3.4}$$

3.4.1 Learning models for the primitives

In order to recognize the primitives we need to have a prototypical representation of each primitive, i.e., a mean and covariance in the 4D feature space. As can be seen in figure 3.2 the actions are all fronto-parallel. Ongoing work aims to generalize this work by allowing for multiple viewpoints. One problem with multiple viewpoints is how to train the system - it will require a very large number of test sequences. Therefore we have captured all training data using a magnetic tracking system with four sensors. The sensors are placed at the wrist, at the elbow, at the shoulder, and at the upper torso (for reference). The hardware used is the Polhemus FastTrac [18] which gives a maximum sampling rate of 25Hz when using four sensors. The data is converted into four Euler angles: three at the shoulder and one at the elbow in order to make the data invariant to body size. An action corresponds to a trajectory through a 4D space spanned by the Euler angles.

The data is input to a commercial computer graphics human model, Poser [19], which then animates all captured data. This allows us to generate training data for any view point and to generate additional training data by varying the Euler angles (based on the training data) and varying the clothing of the model. Figure 3.6 shows a person with magnetic trackers mounted on the arm, two different visualizations of the 3D tracker data from Poser, and an example of the test data. Based on this synthetic training data we build a classifier for each primitive.

Figure 3.6: An illustration of the different types of data used in the system. From left to right: 1) 3D tracker data is acquired from magnetic trackers mounted on persons who perform the five actions. 2) The tracker data is animated in Poser from a fronto-parallel view. 3) The tracker data can be animated from any view point with different clothings and models. 4) After training the primitives on semi-sythetic data we recognize actions in real video.

3.4.2 Defining the primitives

Defining the number of primitives and their characteristics ("human movement") is quite a significant optimization problem. We are aiming at automating this process [17], but in this work the definition of primitives was done manually.

The primitives are defined based on an evaluation of video sequences showing three different people performing the five actions. The criteria for defining the primitives are 1) that they represent characteristic and representative 3D configurations, 2) that their projected 2D configurations contain a certain amount of fronto-parallel motion, and 3) that the primitives are used in the description of as many actions as possible, i.e., fewer primitives are required. In this way we find 10 primitives that can represent the five actions. Each primitive is appearing in several actions resulting in five to eight primitives for each action.

To obtain the prototypical representation we randomly select 20 samples of each primitive and render the appropriate motion capture data to get a computer graphics representation of that sample. The double difference images of these samples are calculated and each of the motion-clouds are represented by the four features. The

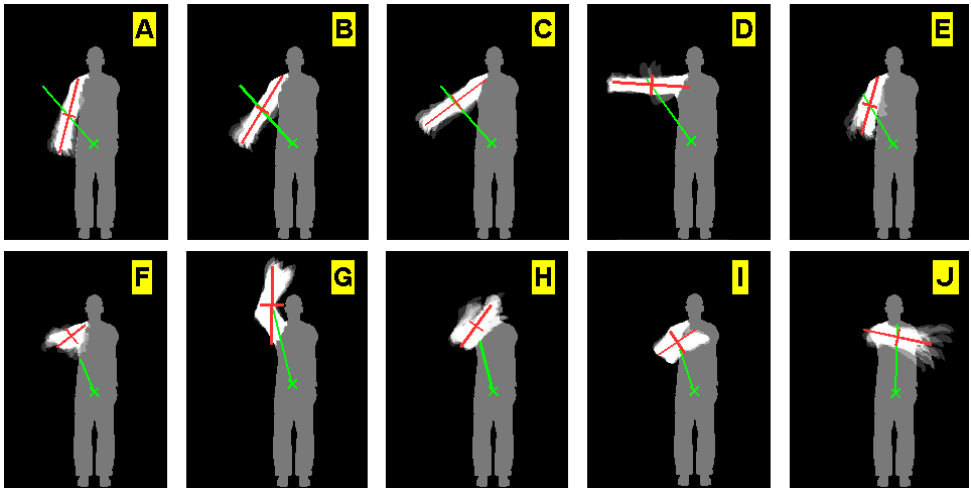


Figure 3.7: The figure of each primitive contains the silhouettes of the 20 samples added together which gives the gray silhouette. The 20 motion clouds from the double difference images of the samples are added on top of the silhouette as the white cloud. The figures furthermore illustrates the mean of the four features for each primitive by depicting the axes of the fitted ellipse and the distance and direction from the reference point to the motion cloud.

20 samples then yields a mean vector and a 4x4 covariance matrix for each primitive. In figure 3.7 the 10 primitives and their representations are visualized together with the letter denoting the primitive. We can use the computer generated version of the training samples in stead of the original real video since the resulting double difference images are comparable and with this approach we achieve the possibility of generating new training data in a fast and flexible way without recording new training video.

3.5 Recognition of actions

The result of recognizing the primitives is a string of letters referring to the known primitives. During a training phase a string representation of each action to be recognized is learned. The task is now to compare each of the learned actions (strings) with the detected string. Since the learned strings and the detected strings (possibly including errors!) will in general not have the same length, the standard pattern recognition methods will not suffice. We therefore apply the Edit Distance method [12], which can handle matching of strings of different lengths.

The edit distance is a well known method for comparing words or text strings, e.g., for spell-checking and plagiarism detection. It operates by measuring the distance

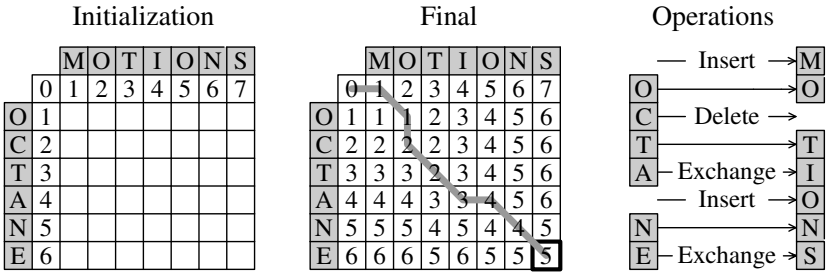


Figure 3.8: Measuring the distance between two strings using edit distance.

between two strings in terms of the number of operations needed in order to transform one into the other. There are three possible operations: *insert* a letter from the other string, *delete* a letter, and *exchange* a letter by one from the other string. Whenever one of these operations is required in order to make the strings more similar, the score or distance is increased. The algorithm is illustrated in figure 3.8 where the strings *motions* and *octane* are compared.

The first step is initialization. The two strings are placed along the sides of the matrix, and increasing numbers are place along the borders beside the strings. Hereafter the matrix is filled cell by cell by traversing one column at a time. If the letters at row i and column j are the same then cell $c_{i,j}$ is assigned the same value as cell $c_{i-1,j-1}$. Otherwise cell $c_{i,j}$ is assigned the smallest value of the following three operations:

$$Insert: \quad c_{i-1,j} + cost \tag{3.5}$$

$$Delete: \quad c_{i,j-1} + cost \tag{3.6}$$

$$Exchange: \quad c_{i-1,j-1} + cost \tag{3.7}$$

In the original edit distance method the *cost* equals one.

Using these rules the matrix is filled and the value found at the bottom right corner is the edit distance required in order to map one string into the other, i.e., the distance between the two strings. The actual sequence of operations can be found by back-tracing the matrix. More than one path is often possible.

The edit distance is a deterministic method but by changing the cost of each of the three operations with respect to likelihoods it becomes a probabilistic method. The edit distance method has several variations that define cost functions in different ways, e.g. the Weighted Edit Distance where a cost function is defined for each operation or the Needleman-Wunsch algorithm [14] where a cost matrix is used to define the similarity between the symbols (letters) of the applied set of symbols (alphabet). In stead of defining a fixed cost for an operation or each symbol-pair we define the cost of applying operations to a primitive based on the actual observations at any given time. The number of repetitions of a primitive to some extent represent the likelihood of that primitive being correct. This means, in terms of a cost function, that the cost of deleting or exchanging a primitive that have been observed multiple

times should be increased with the number of observed repetitions. Concretely we incorporate the weights described above (see equation 3.4) into the edit distance method by increasing the cost of the *delete* and *exchange* operations by the weight multiplied by β (a scaling factor). The cost of *inserting* remains 1.

When the strings representing the actions are of different lengths, the method tends to favor the shorter strings. Say we have detected the string $\{B, C, D\}$ and want to classify it as being one of the two actions: $a_1 = \{J, C, G\}$ and $a_2 = \{A, B, C, D, H\}$. The edit distance from the detected string to the action-strings will be two in both cases. However, it seems more likely that the correct interpretation is that the detected string comes from a_2 in a situation where the start and end has been corrupted by noise. In fact, 2 out of 3 of the primitives have to be changed for a_1 whereas only 2 out of 5 have to be changed for a_2. We therefore normalize the edit distance by dividing the output by the length of the action-string, yielding 0.67 for a_1 and 0.2 for a_2, i.e., a_2 is recognized.

The above principle works for situations where the input sequence only contains one action (possibly corrupted by noise). In a real scenario, however, we will have sequences which are potentially much longer than an action and which might include more actions after each other. The action recognition problem is therefore formulated as for each action to find the substring in the detected string, which has the minimum edit distance. The recognized action will then be the action that has the substring with the overall minimum edit distance. Denoting the start point and length of the substring, s and l, respectively, we recognize the action present in the detected string as:

$$\text{Action} = \arg\min_{k,s,l} PED(\Lambda, k, s, l) \tag{3.8}$$

where k index the different actions, Λ is the detected string, and $PED(\cdot)$ is the probabilistic edit distance.

3.6 Results

3.6.1 Test setup

Two kinds of tests are conducted: one with known start and stop time of action execution, and another with "noise" added in the beginning and end of the sequences (unknown start time). By adding noise to the sequence we introduce the realistic problem of having no clear idea about when an action commences and terminates which would be the case in a real situation. To achieve a test scenario that resembles this situation we split the five actions into halves and add one of these half actions to the beginning and one to the end of each action to be processed by the system. The added half actions are chosen randomly resulting in unknown start and end point of the real action.

We use eleven test subjects, whom each performs each gesture 10 times. This leads to 550 sequences. The weighting of the features $\vec{W}$ are set to $\{1, 4, 2, 4\}$, and $\beta = 1/8$.

$\vec{W}$ and β are determined through quantitative tests. A string representation of each action is found and since the shortest string contains five primitives and the longest eight primitives, we only perform the probabilistic edit distance calculation for substrings having the lengths $\in [3, 15]$.

3.6.2 Tests

The overall recognition rate for the test with known start time is 88.7%. In figure 3.9(a) the confusion matrix for the results is shown. As can be seen in the figure, most of the errors occur by miss-classification between the two actions: *move closer* and *raise arm*. The main reasons for this confusion are the similarity of the actions, the similarity of the primitives in these actions, and different performances of the actions of different test subjects (some do not raise their arm much when preforming the *raise arm* action). As can be seen in figure 3.2 both actions are performed along the side of the person when seen from the fronto-parallel view and differs mainly in how high the arm is raised. From figure 3.7 it can be seen that primitives 'F', 'G', 'H', and 'I' have similar angles between the reference point and the motion cloud and 'F', 'H' and 'I' also have similar orientation of the ellipse. These two features, which are the ones with highest weights, make these four primitives harder to distinguish.

Figure 3.9(b) shows the confusion matrix for the test results with noise. The overall recognition rate for this test is 85.5%. The errors are the same as before but with some few additional errors caused by the unknown start and end time of the actions.

	1.	2.	3.	4.	5.
1. Point right	**100**				
2. Move left	**6.4**	**90.9**		**2.7**	
3. Move right	**5.5**		**92.7**	**0.9**	**0.9**
4. Move closer		**2.7**	**1.8**	**70.9**	**23.6**
5. Raise arm				**10.9**	**89.1**

(a) Known start and stop time.

	1.	2.	3.	4.	5.
1. Point right	**99.1**		**0.9**		
2. Move left	**9.1**	**90.0**		**0.9**	
3. Move right	**7.3**		**90.0**	**2.7**	
4. Move closer	**0.9**	**4.5**	**1.8**	**62.7**	**30.0**
5. Raise arm	**1.8**	**1.8**		**10.9**	**85.5**

(b) Unknown start and stop time.

Figure 3.9: The confusion matrices for the recognition rates (in percent) without added noise (a) and with added noise (b). Zero values have been left out to ease the overview of the confusion.

3.7 Conclusion

In this paper we have presented an action recognition approach based on motion primitives as opposed to trajectories. Furthermore, we extract features from temporally local motion as opposed to background subtraction or another segmentation

method relying on learned models and a relatively controlled environment. We hope this makes our approach less sensitive, but have still to prove so in a more comprehensive test.

The results are promising due to two facts. First, the models are generated from synthetic data (generated based on test subjects) while the test data are real data. In fact, the test data and training data are recorded several months apart, hence this is a real test of the generalization capabilities of the action recognition process. This means that we can expect to use the same scheme when learning models for the next incarnation of the system, which is aimed at view-invariant action recognition. Secondly, the system does not break down when exposed to realistic noise. This suggests that the approach taking has potential to be expanded into a real system setup, as opposed to a lab setup which is virtually always used when testing action recognition systems.

The primitives used in this work are found manually. This turned out to be quite an effort due to the massive amount of data and possibilities. Currently we are therefore working to automate this process [17]. Another ongoing activity is to avoid manually defining the reference point, see section 3.3, by using the face as a reference for the features [13].

References

[1] R.V. Babu and K.R. Ramakrishnan. Compressed domain human motion recognition using motion history information. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing"*, Hong Kong, April, 2003.

[2] J. Barbic, N.S. Pollard, J.K. Hodgins, C. Faloutsos, J-Y. Pan, and A. Safonova. Segmenting Motion Capture Data into Distinct Behaviors. In *Graphics Interface*, London, Ontario, Canada, May 17-19 2004.

[3] F. Bettinger and T.F. Cootes. A Model of Facial Behaviour. In *IEEE International Conference on Automatic Face and Gesture Recognition*, Seoul, Korea, May 17 - 19 2004.

[4] A. Bobick and J. Davis. The Recognition of Human Movement Using Temporal Templates. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.

[5] A.F. Bobick and J. Davis. A Statebased Approach to the Representation and Recognition of Gestures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(12):1325 – 1337, 1997.

[6] C. Bregler. Learning and Recognizing Human Dynamics in Video Sequences. In *Conference on Computer Vision and Pattern Recognition*, pages 568 – 574, San Juan, Puerto Rico, 1997.

[7] L. Campbell and A. Bobick. Recognition of Human Body Motion Using Phase Space Constraints. In *International Conference on Computer Vision*, Cambridge, Massachusetts, 1995.

[8] J. Gonzalez, J. Varona, F.X. Roca, and J.J. Villanueva. *aSpaces*: Action spaces for recognition and synthesis of human actions. In *AMDO*, pages 189–200, 2002.

[9] O.C. Jenkins and M.J. Mataric. Deriving Action and Behavior Primitives from Human Motion Data. In *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, pages 2551–2556, Lausanne, Switzerland, Sept., 2002.

[10] A. Just and S. Marcel. HMM and IOHMM for the Recognition of Mono- and Bi-Manual 3D Hand Gestures. In *ICPR workshop on Visual Observation of Deictic Gestures (POINTING04)*, Cambridge, UK, August 2004.

[11] A. Kale, N. Cuntoor, and R. Chellappa. A Framework for Activity-Specific Human Recognition. In *International Conference on Acoustics, Speech and Signal Processing*, Orlando, Florida, May 2002.

[12] V.I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848, 1965.

[13] T.B. Moeslund, J.S. Petersen, and L.D. Skalski. Face Detection Using Multiple Cues. In *Scandinavian Conference on Image Analysis*, Aalborg, Denmark, June 10-14 2007.

[14] S.B. Needleman and C.D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–53, 1970.

[15] Alonso Patron-Perez and I. Reid. A Probabilistic Framework for Recognizing Similar Actions using Spatio-Temporal Features. In *British Machine Vision Conference*, UK, Sep. 2007.

[16] C. Rao, A. Yilmaz, and M. Shah. View-Invariant Representation and Recognition of Actions. *Journal of Computer Vision*, 50(2):55 – 63, 2002.

[17] L. Reng, T.B. Moeslund, and E. Granum. Finding Motion Primitives in Human Body Gestures. In S. Gibet, N. Courty, and J.-F. Kamps, editors, *GW 2005*, number 3881 in LNAI, pages 133–144. Springer Berlin Heidelberg, 2006.

[18] *http://polhemus.com/*, January 2006.

[19] *http://www.poserworld.com/*, January 2006.

[20] D. Weinland, R. Ronfard, and E. Boyer. Free Viewpoint Action Recognition using Motion History Volumes. *Computer Vision and Image Understanding*, 104(2):249–257, 2006.

[21] A. Yilmaz and M. Shah. Actions Sketch: A Novel Action Representation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, CA, June, 2005.

[22] K. Yoshinari and M. Michihito. A Human Motion Estimation Method using 3-Successive Video Frames. In *Int. Conf. on Virtual Systems and Multimedia*, Gifu, Japan, 1996.

[23] H. Yu, G.-M. Sun, W.-X. Song, and X. Li. Human Motion Recognition Based on Neural Networks. In *ICCCS*, Hong Kong, May 2005.

Chapter 4

3D Human Gesture Recognition

This chapter consists of the paper "View-Invariant Gesture Recognition using 3D Optical Flow and Harmonic Motion Context" [A]. The paper presents work that builds on the gesture recognition of chapter 3. The principle of motion primitives for gesture recognition is in this paper used to develop a view invariant method by use of a 3D camera. The paper is presented here to document a direct and very interesting extension of the gesture recognition by motion primitives. The first author of the paper is however to be credited for most of the work in this publication.

References

A. M.B. Holte, T.B. Moeslund and P. Fihl. View-Invariant Gesture Recognition using 3D Optical Flow and Harmonic Motion Context. In *Computer Vision and Image Understanding*, December 2010

View-Invariant Gesture Recognition using 3D Optical Flow and Harmonic Motion Context

M.B. Holte, T.B. Moeslund and P. Fihl

Abstract

This paper presents an approach for view-invariant gesture recognition. The approach is based on 3D data captured by a SwissRanger SR4000 camera. This camera produces both a depth map as well as an intensity image of a scene. Since the two information types are aligned, we can use the intensity image to define a region of interest for the relevant 3D data. This data fusion improves the quality of the motion detection and hence results in better recognition. The gesture recognition is based on finding motion primitives (temporal instances) in the 3D data. Motion is detected by a 3D version of optical flow and results in velocity annotated point clouds. The 3D motion primitives are represented efficiently by introducing motion context. The motion context is transformed into a view-invariant representation using spherical harmonic basis functions, yielding a harmonic motion context representation. A probabilistic Edit Distance classifier is applied to identify which gesture best describes a string of primitives. The approach is trained on data from one viewpoint and tested on data from a very different viewpoint. The recognition rate is 94.4% which is similar to the recognition rate when training and testing on gestures from the same viewpoint, hence the approach is indeed view-invariant.

4.1 Introduction

Automatic analysis of humans and their actions has received increasingly more attention in the last decade [20]. One of the areas of interest is recognition of human gestures for use in for example Human Computer Interaction.

Many different approaches to gesture recognition have been reported [19]. They apply a number of different segmentation, feature extraction, and recognition strategies. E.g. [25] and [28] extract and represent human gestures/actions by velocity histories of tracked keypoints and ballistic dynamics, respectively, while gestures are recognized, e.g., through Hidden Marcov Models (HMMs) [2, 26, 27] or Dynamic Baysian Networks (DBNs) [3, 35]. These methods are virtually all based on analyzing 2D data, i.e., images. A consequence of this is that approaches only analyze 2D gestures carried out in the image plane, which is only a projection of the actual gesture. As a result, the projection of the gesture will be dependent on the viewpoint, and not contain full information about the performed gesture. To overcome this shortcoming the use of 3D data has been introduced through the use of two or more cameras, see for example [5, 7]. In this way, e.g., the surface structure or a 3D volume of the person can be reconstructed, and thereby a more descriptive representation for gesture recognition can be established. We follow this line of work and also apply 3D data. To avoid the difficulties inherent to classical stereo approaches (the correspondence problem, careful camera placement and calibration) we instead apply a *Time-of-Flight* (ToF) range camera – SwissRanger SR4000. Each pixel in this camera directly provides a depth value (distance to object). Even though the technology in range cameras is still in its early days, e.g., resulting in low resolution data, the great potential of such sensors has already resulted in them being applied in a number of typical computer vision applications like face detection [6], face tracking [17], shape analysis [13, 21], robot navigation [24] and gesture-based scene navigation [29]. In [1] a survey of recent developments in ToF-technology are presented. It discusses applications of this technology for vision, graphics, and HCI.

The development of range cameras has progressed rapidly over the last few years, leading to the release of new and improved camera models from some of the main manufacturers: MESA Imaging [18], PMD Technologies [23] and 3DV Systems [33]. Recently, MESA Imaging released the new SwissRanger SR4000 range camera with higher frame rate (up to 54 fps) and resolution (176×144 pixels). 3DV Systems is aiming at a consumer class range camera with similar size and look as a regular web-camera and a integrated sensor capable of producing 1 mega pixels color images, while PMD Technologies made a camera version with improved operating range (up to 40 m) for e.g. pedestrian detection in cars.

The SwissRanger camera that we apply also provides an amplitude value corresponding to an intensity value for each pixel. This means that at any given time instant both a depth image and an intensity image are present. For some applications these two information types compliment each other and are therefore both used. For example in [21] where the objective is to segment planar surfaces in 3D (range) data, the edges in the intensity image are applied to improve the result. Similar benefits

of applying both data types can be seen in [6, 17, 8]. We also apply both data types and will show how they compliment each other.

Applying 3D data allows for analysis of 3D gestures. However, we are still faced with the problem that a user has to be fronto-parallel with respect to the camera. A few works have been reported without the assumption on the user being fronto-parallel. E.g. in [30] where 5 calibrated and synchronized cameras are used to acquire data (the publicly available IXMAS data set), which is further projected to 64 evenly spaced virtual cameras used for training. Actions are described in a view-invariant manner by computing $\mathcal{R}$ transform surfaces and manifold learning. Similarly, [7] use the same data set to compute motion history volumes, which are used to derive view-invariant motion descriptors in Fourier space. Another example is seen in [5] where 3D Human Body Shapes are used for view-independent identification of human body postures, which are trained and tested on another multi-camera dataset.

The need for multiple calibrated and synchronized cameras followed up by an exhaustive training phase for multiple viewpoints is obviously not desirable. Instead we aim at a view-invariant approach which is trained by examples from one camera viewpoint and able to recognize gestures from a very different viewpoint, say $\pm 45°$. Another issue we want to combat is the often used assumption of known start- and end points. That is, often the test data consists of N sequences where each sequence contains one and only one gesture. This obviously makes the problem easier and it favors a trajectory-based approach, where each gesture is represented as a trajectory though some state-space with known start and end point. For real-life scenarios the start and end point is normally not known. To deal with this issue we follow the notion of recognition through a set of primitives [10, 11, 34, 37]. Concretely, we define a primitive as a time instance with significant 3D motion.

4.1.1 Our approach

So, we represent gestures as an ordered sequence of *3D motion primitives* (temporal instances). We focus on arm gestures and therefore only segment the arms (when they move) and hereby suppress the rest of the (irrelevant) body information. Concretely we extract the moving arms using a 3D version of *optical flow* to produce *velocity annotated point clouds* and represent this data efficiently by their *motion context*. The motion context is an extended version of the regular shape context [4], and represents the velocity annotated point cloud by using both the location of motion, together with the amount of motion and its direction. We make the primitives invariant to rotation around the vertical axis by re-representing the motion context using *spherical harmonic basis functions*, yielding a *harmonic motion context* representation. In each frame the primitive, if any, which best explains the observed data is identified. This leads to a discrete recognition problem since a video sequence of range data will be converted into a string containing a sequence of symbols, each representing a primitive. After pruning the string a *probabilistic Edit Distance classifier* is applied to identify which gesture best describes the pruned string. Our approach is illustrated in Figure 4.1.

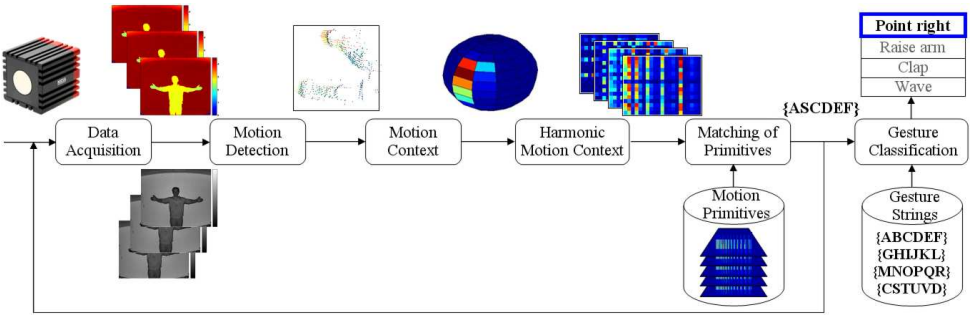


Figure 4.1: An overview of the range and intensity based gesture recognition system. Note that the feedback loop illustrates that a number of frames are processed before recognition of gestures commences.

4.1.2 Structure of the paper

This paper is organized as follows. Data acquisition and preprocessing is presented in Section 4.2, followed up by how we perform motion detection in 3D. In Section 4.3 we describe the concept of motion primitives, and how they are represented compactly by introducing motion context. Furthermore, we show how the motion context can be transformed into a view-invariant representation using spherical harmonic basis functions, yielding a harmonic motion context representation. In Section 4.4 we describe the classification of motion primitives, and how we perform gesture recognition by introducing a probabilistic edit distance classifier. Finally, we present experimental results in Section 4.5 and concluding remarks in Section 4.6.

4.2 Segmentation

4.2.1 Data acquisition and preprocessing

We capture intensity and range data using a SwissRanger SR4000 range camera from MESA Imaging. The camera is based on the Time-of-Flight (ToF) principle and emits radio-frequency modulated (30 MHz) light in the near-infrared spectrum (850 nm), which is backscattered by the scene and detected by a CMOS CCD. The resolution is 176×144 pixels with an active range of 0.3 to 5.0 meters and a field of view of $43.6° \times 34.6°$. The distance accuracy is typically in the order of ± 1 centimeter, depending of the distance range and illumination. Figure 4.2 shows an intensity and a range image of one time instant of a "clap" gesture.

Due to the different reflection properties of materials and the light condition in a captured scene, scattering effects of the active illumination emitted by the camera occurs, hence some noise will be present in the data. To deal with these noise effects we have applied a number of preprocessing techniques proposed in [31]. The preprocessing consists of the following steps: *smoothing of the distance images with a*

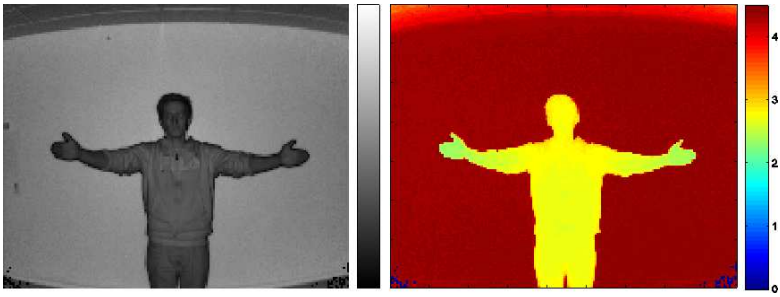


Figure 4.2: An intensity and a range image produced by the SwissRanger SR4000 camera, where the pixel values correspond to a distance in the range image.

distance-adaptive median filter, *thresholding on the amplitude values*, and *edge point removal*. Especially, the removal of edge points is of high importance in the case of gesture recognition. Edge points arise in the case where reflected light from the foreground and the background hits the same pixel simultaneously. The camera sensor is controlled as a so-called 1-tap sensor. This means that in order to obtain distance information, four consecutive exposures have to be performed [22]. If the distance a pixel "sees" changes in this time window, the distance calculation is falsified. The measurement returned by the pixel will be a falsified distance somewhere between the foreground and the background. Fast moving objects in the scene may therefore cause errors in the distance calculations. This error is inversely proportional to the frame rate. The problem especially occurs in regions of a scene that contain objects with high velocity and distance gradients; in our case a fast moving arm. The pixel could then see the arm for the first two taps and a wall for the last two. Thus, the edges of the moving arm are poorly defined and lead to incorrect distance measurement. Concretely, when visualizing the range data as a 3D point cloud, the points origin from these regions are "stretching" backwards along the edges of the moving arm. Since we are interested in gesture recognition where a lot of motion is obviously present, the ToF data can easily be corrupted by a significant amount of neighboring edge points.

4.2.2 3D motion detection

We detect movements (of the arms) using a 3D version of optical flow to produce velocity annotated point clouds [32]. Optical flow is the pattern of apparent motion in a visual scene caused by the relative motion between an observer and the scene. The main benefit of optical flow compared to other motion detection techniques, like double differencing [36] which we have used in an earlier versions of this work [8], is that optical flow determines both the amount of motion and its direction in form of velocity vectors. The following description of the motion detection is inspired by [32] and to some extent quoted or paraphrased. However, the full description along with some additional information and comments are included in this section, as this

is an important part of our approach and for the convenience of the reader.

The technique computes the optical flow of each image pixel as the distribution of apparent velocity of moving brightness patterns in an image. The flow of a constant brightness profile can be described by the constant velocity vector $\mathbf{v}_{2D} = (v_x, v_y)^T$ as outlined in Equation 4.1.

$$\begin{aligned} I(x,y,t) &= I(x+\delta x, y+\delta y, t+\delta t) \\ &= I(x+v_x\cdot\delta t, y+v_y\cdot\delta t, t+\delta t) \\ \Rightarrow\ & \frac{\partial I}{\partial x}\cdot v_x + \frac{\partial I}{\partial y}\cdot v_y = -\frac{\partial I}{\partial t} \end{aligned} \tag{4.1}$$

Usually, the estimation of optical flow is based on differential methods. They can be classified into global strategies which attempt to minimize a global energy functional [9] and local methods, that optimize some local energy-like expression. A prominent local optical flow algorithm developed by Lucan and Kanade [16] uses the spatial intensity gradient of an image to find matching candidates using a type of Newton-Raphson iteration. They assume the optical flow to be constant within a certain neighborhood, which allows to solve the optical flow constraint equation (Equation 4.1) via least square minimization.

A characteristic of the Lucas-Kanade algorithm, and that of other local optical flow algorithms, is that it does not yield a very high density of flow vectors, i.e. the flow information fades out quickly across motion boundaries and the inner parts of large homogenous areas show little motion. However, its advantage is the comparative robustness in presence of noise. In the case of the data obtained by ToF cameras, with low resolution and often affected by a high amount of noise, this is a very important property. We use a hierarchical implementation of the Lucas-Kanade algorithm [12] which has successfully been applied in [32].

The optical flow is computed for each frame $\mathcal{F}_i$ of a sequence of intensity images provided by the SwissRanger camera $(\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_n)$ and based on data from two consecutive frames $(\mathcal{F}_i, \mathcal{F}_{i-1})$. Each pixel of frame $\mathcal{F}_i$ is annotated with a 2D velocity vector $\mathbf{v}_{2D} = (v_x, v_y)^T$ as shown in Figure 4.3, which results in pixel correspondences between frame $\mathcal{F}_i$ and frame $\mathcal{F}_{i-1}$. As a 3D point is available for each pixel these pixel correspondences can be directly transformed into 3D point correspondences $(\mathbf{p}_k^i, \mathbf{p}_l^{i-1})$ which can be used to compute 3D velocities $\mathbf{v}_{3D} = (v_x, v_y, v_z)^T = \mathbf{p}_k^i - \mathbf{p}_l^{i-1}$. Figure 4.4 presents multiple viewpoints of a 3D point cloud of a time instant in a sequence annotated with 3D velocity vectors. In Figure 4.4 (left) the moving arms are present in the data, but so is a large amount of noise due to erroneous depth values often produced by the SwissRanger camera. Furthermore, points origin from most of the human body is present due to small movements or deviations in the distance measurements. These insignificant and erroneous velocity vectors are eliminated to some extent by simple filtering and thresholding as shown in Figure 4.4 (right).

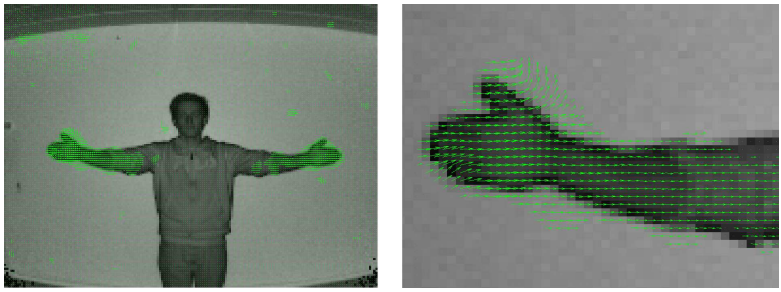


Figure 4.3: An input image overlaid with the estimated 2D optical flow vectors.

4.3 Motion primitives

4.3.1 Motion context

After motion detection we are left with a velocity annotated point cloud in 3D, which is represented efficiently using a motion oriented version of shape context. We call this representation the *motion context*.

A shape context [4] is based on a spherical histogram. This histogram is centered in a reference point and divided linearly into S azimuthal (east-west) bins and T colatitudinal (north-south) bins, while the radial direction is divided into U bins. Figure 4.5 gives an example of the shape context descriptor.

Usually, the value of a bin is given by the number of 3D points falling within that particular bin. However, the motion context extends the regular shape context to represent the velocity annotated point cloud by using both the location of motion, together with the amount of motion and its direction. For each bin in the shape context we accumulate the annotated velocity vectors, of each 3D point falling within that particular bin, into an orientation histogram. Specifically, we introduce a Histogram of Optical Flow (HOF). The idea of HOF is the same as in the Histogram of Oriented Gradients (HOG) used in the popular Scale Invariant Feature Transform (SIFT) [15]. However, we extend this to 3D and in contrast to use gradient vectors, we use velocity vectors. We divide the HOF representation into s azimuthal (east-west) orientation bins and t colatitudinal (north-south) bins, where each bin is weighted by the length of the velocity vectors falling within the bin. This results in a $S \times T \times U \times s \times t$ dimensional feature vector for each frame. The HOF representation and how it is divided into azimuthal and colatitudinal bins is illustrated in Figure 4.6.

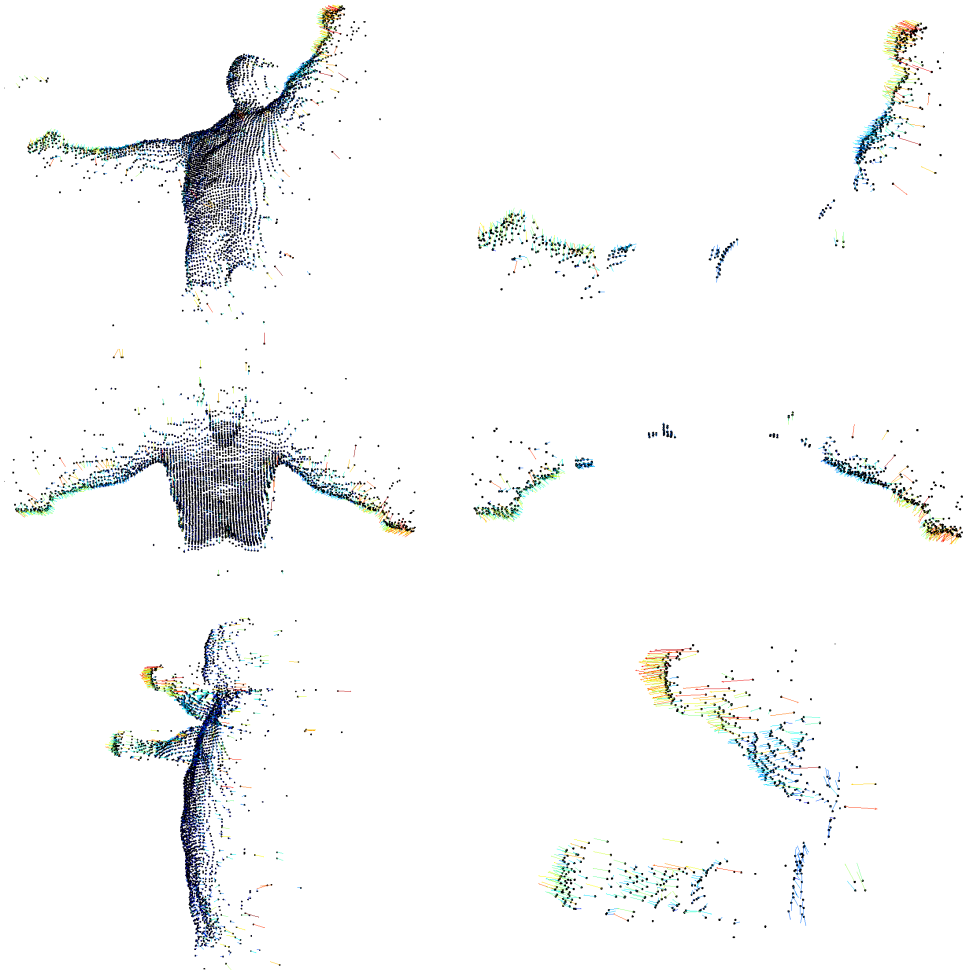


Figure 4.4: Left: Three different viewpoints of a 3D point cloud annotated with 3D velocity vectors. The data has been extracted from a time instant of a "clap" gesture. Right: The same three viewpoints of the velocity annotated point cloud after filtering and thresholding to remove insignificant and erroneous velocity vectors. The points are illustrated with black dots while the velocity vectors are color coded. Blue represents a low velocity while red represents a high velocity. Note: the scale of the sub-figures varies for illustrative purpose.

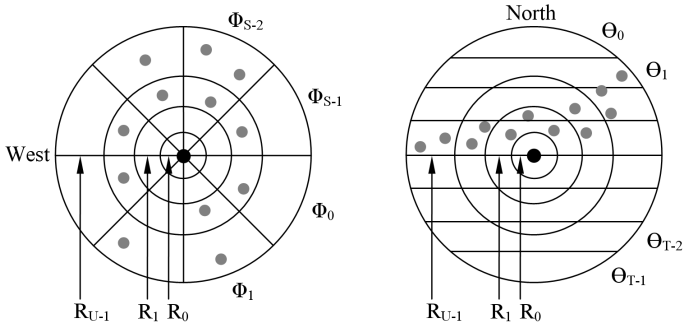


Figure 4.5: A horizontal and a vertical cross-section of a shape context descriptor.

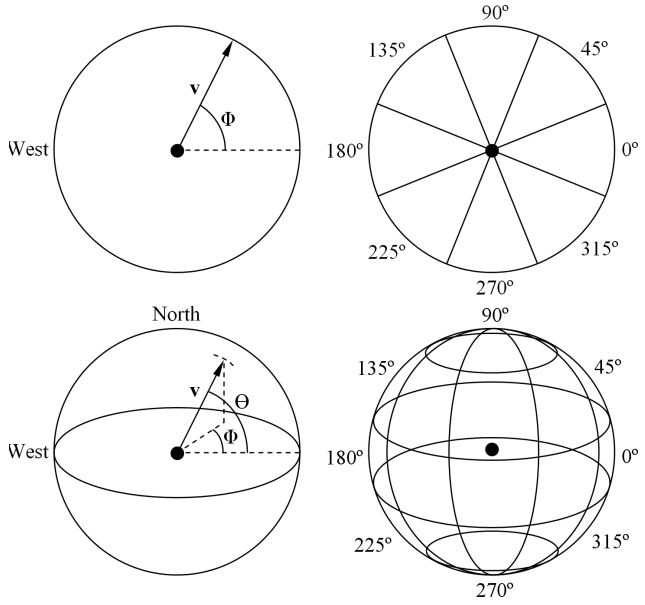


Figure 4.6: The HOF descriptor and how it is divided into 8 azimuthal and 4 colatitudinal bins illustrated by a top-view in 2D and 3D.

In SIFT, partially illumination invariance is imposed by thresholding and normalizing the feature vector. In the same way we impose partially invariance towards the velocity of movements, like in the case where two individuals perform the same gesture at different speed. Hence, the feature vector will have greater emphasis to the location and orientation, while reducing the influence of large velocity values.

4.3.2 View-invariant representation: harmonic motion context

By introducing spherical harmonics we can eliminate one of the two rotational parameters in a shape context descriptor. Similarly, the motion context descriptor can be transformed by using this technique first for each of the HOF descriptors, and thereafter for the entire motion context representation. We eliminate the rotation around the vertical axis, see Figure 4.5 and 4.6, and hereby make our representation invariant to variations in this parameter.

Any given spherical function, i.e. a function $f(\theta, \phi)$ defined on the surface of a sphere parameterized by the colatitudinal and azimuthal variables θ and ϕ, can be decomposed into a weighted sum of spherical harmonics as given by Equation 4.2.

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} A_l^m \, Y_l^m(\theta, \phi) \tag{4.2}$$

The term A_l^m is the weighing coefficient of *degree* m and *order* l, while the complex functions $Y_l^m(\cdot)$ are the actual spherical harmonic functions of *degree* m and *order* l. In Figure 4.7 some examples of higher order spherical harmonic basis functions are illustrated.

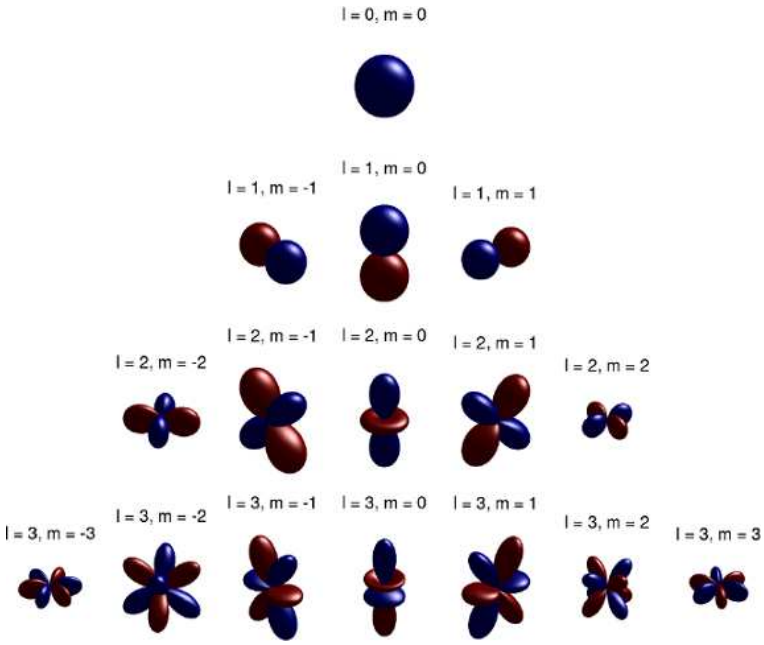


Figure 4.7: Illustration of some higher order spherical harmonic basis functions of degree m and order l. Blue indicates positive values and red negative values.

The following states the key advantages of the mathematical transform based on the family of orthogonal basis functions in the form of spherical harmonics. The complex function $Y_l^m(\cdot)$ is given by Equation 4.3.

$$Y_l^m(\theta,\phi) = K_l^m \, P_l^{|m|}(\cos\theta) \, e^{jm\phi} \tag{4.3}$$

The term K_l^m is a normalization constant, while the function $P_l^{|m|}(\cdot)$ is the *associated Legendre Polynomial*. The key feature to note from Equation 4.3 is the encoding of the azimuthal variable ϕ. The azimuthal variable solely inflects the *phase* of the spherical harmonic function and has no effect on the *magnitude*. This effectively means that $||A_l^m||$, i.e. the norm of the decomposition coefficients of Equation 4.2 is invariant to parameterization in the variable ϕ.

The actual determination of the spherical harmonic coefficients is based on an inverse summation as given by Equation 4.4, where N is the number of samples ($S \times T$). The normalization constant $4\pi/N$ originates from the fact, that Equation 4.4 is a discretization of a continuous double integral in spherical coordinates, i.e. $4\pi/N$ is the surface area of each sample on the unit sphere.

$$(A_l^m)_{f_u} = \frac{4\pi}{N} \sum_{\phi=0}^{2\pi} \sum_{\theta=0}^{\pi} f_u(\theta,\phi) \; Y_l^m(\theta,\phi) \tag{4.4}$$

In a practical application it is not necessary (or possible, as there are infinitely many) to keep all coefficient A_l^m. Contrary, it is assumed the functions f_u (f_u are the spherical functions for each of the given spherical shells $u \in [0; U-1]$) are band-limited, hence it is only necessary to keep coefficient up to some bandwidth $l = B$.

The band-limit assumption effectively means, that each spherical shell is decomposed into $(B+1)^2$ coefficients (i.e., the number of terms in the summation $\sum_{l=0}^{B}\sum_{m=-l}^{l}$ in Equation 4.2). By using the fact, that $||A_l^m|| = ||A_l^{-m}||$ and only saving coefficients for $m \geq 0$, the number of describing coefficients for each spherical shell is reduced to $(B+1)(B+2)/2$ coefficients (i.e., the number of terms in the summation $\sum_{l=0}^{B}\sum_{m=0}^{l}$). Given the U different spherical shells, the dimensionality of the feature vector becomes $D = U(B+1)(B+2)/2$.

However, since each bin of the spherical motion context representation consists of an embedded spherical function in form of a HOF representation, we first transform each of the inner HOF representations up to some bandwidth B_1, and thereafter we transform the entire motion context up to some bandwidth B_2. Hence, the dimensionality of each transformed HOF representation D_1 and the transformed motion context D_2 becomes $D_1 = (B_1+1)(B_1+2)/2$ and $D_2 = U(B_2+1)(B_2+2)/2$, respectively. When the HOF representations have been transformed, each cell in the motion context consists of an array of spherical harmonic coefficients. This means that the second transformation has to be done with respect to these coefficients, hence the resulting dimensionality of the final feature vector becomes

$$D = D_1 D_2 = U(B_1+1)(B_1+2)(B_2+1)(B_2+2)/4 \tag{4.5}$$

Concretely we set $U = 4$, $B_1 = 4$ and $B_2 = 5$, resulting in 4×315 coefficients (see Figure 4.8).

The spherical motion context histogram is centered in a reference point, which is estimated as the center of gravity of the human body, and the radial division into U bins is made in steps of 25 cm. Furthermore, we set $S = 12$, $T = 6$, $s = 8$ and $t = 4$.

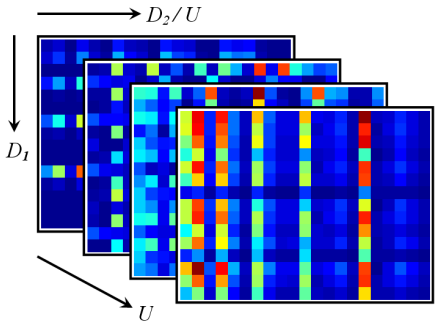


Figure 4.8: An example of a harmonic motion context representation. Where each of the 4 stacked images shows the spherical harmonic coefficients for the 4 radial divisions in the motion context, respectively.

4.4 Classification

The classification is divided into two main tasks: recognition of motion primitives by use of the harmonic motion context descriptors, and recognition of the actual gestures using an ordered sequence of primitives (see Figure 4.1).

4.4.1 Recognition of primitives: correlation

A motion primitive is recognized by matching the current harmonic motion context with a known set, one for each possible primitive. The actual comparison of two harmonic motion contexts is done by the normalized correlation coefficient as given by Equation 4.6. To this end each harmonic motion context is represented as a vector $\mathbf{h_1}$ and $\mathbf{h_2}$ of length n containing the (stacked) spherical harmonic coefficients for a specific primitive at time, t:

$$match(\mathbf{h_1}, \mathbf{h_2}, t) = \frac{n \sum \mathbf{h_1}\mathbf{h_2} - \sum \mathbf{h_1} \sum \mathbf{h_2}}{\sqrt{\left[n \sum (\mathbf{h_1})^2 - (\sum \mathbf{h_1})^2\right]\left[n \sum (\mathbf{h_2})^2 - (\sum \mathbf{h_2})^2\right]}} \tag{4.6}$$

The system is trained by generating a representative set of descriptors for each primitive. A reference descriptor is then estimated as the average of all these descriptors for each class (primitive).

The classification of a sequence can be viewed as a trajectory through the feature space where, at each time-step, the closest primitive (in terms of the normalized correlation coefficient) is found. To reduce noise in this process we introduce a minimum coefficient in order for a primitive to be considered in the first place. Furthermore, to reduce the flickering observed when the trajectory passes through a border region between two primitives we introduce a hysteresis threshold. It favors the primitive recognized in the preceding frame over all other primitives by modifying the individual distances. The classifier hereby obtains a "sticky" effect, which handles a large part of the flickering.

After processing a sequence the output will be a string with the same length as the sequence. An example is illustrated in Equation 4.7. Each letter corresponds to a recognized primitive and Ø corresponds to time instances where no primitives are detected. The string is pruned by first removing 'Ø's, isolated instances, and then all repeated letters, see Equation 4.8. A weight is generated to reflect the number of repeated letters (this is used below).

$$\text{String} = \{Ø, Ø, B, B, B, B, B, E, A, A, F, F, F, F, Ø, D, D, G, G, G, G, Ø\} \quad (4.7)$$

$$\text{String} = \{B, A, F, D, G\} \quad (4.8)$$

$$\text{Weights} = \{5, 2, 4, 2, 4\} \quad (4.9)$$

4.4.2 Recognition of gestures: probabilistic edit distance

The result of recognizing the primitives is a string of letters referring to the known primitives. During a training phase a string representation of each gesture to be recognized is learned. The task is now to compare each of the learned gestures (strings) with the detected string. Since the learned strings and the detected string (possibly including errors!) will in general not have the same length, the standard pattern recognition methods will not suffice. We therefore apply the Edit Distance method [14], which can handle matching of strings of different lengths.

The edit distance is a well known method for comparing words or text strings, e.g., for spell-checking and plagiarism detection. It operates by measuring the distance between two strings in terms of the number of operations needed in order to transform one to the other. There are three possible operations: *insert* a letter from the other string, *delete* a letter, and *exchange* a letter by one from the other string. Whenever one of these operations is required in order to make the strings more similar, the score or distance is increased by one. The algorithm is illustrated in Figure 4.9 where the strings *motions* and *octane* are compared.

The first step is initialization. The two strings are placed along the sides of the matrix, and increasing numbers are placed along the borders beside the strings. Hereafter the matrix is filled cell by cell by traversing one column at a time. Each cell is given the smallest value of the following four operations:

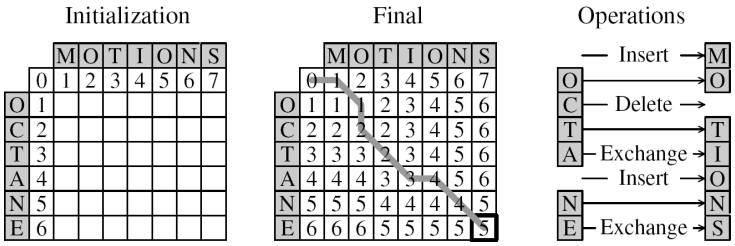


Figure 4.9: Measuring the distance between two strings using edit distance.

Insert: The value of the cell above + 1

Delete: The value of the cell to the left + 1

Exchange: The value of the cell up-left + 1

No change: The value of the cell up-left + 0. This is the case when the letters in question in the two stings are the same.

Using these rules the matrix is filled and the value found at the bottom right corner is the edit distance required in order to map one string into the other, i.e., the distance between the two strings. The actual sequence of operations can be found by back-tracing the matrix. Note that often more paths are possible.

When the strings representing the gestures are of different lengths, the method tends to favor the shorter strings. Say we have detected the string $\{B, C, D\}$ and want to classify it as being one of the two gestures: $\#1 = \{J, C, G\}$ and $\#2 = \{A, B, C, D, H\}$. The edit distance from the detected string to the gesture-strings will be two in both cases. However, it seems more likely that the correct interpretation is that the detected string comes from gesture #2 in a situation where the start and end has been corrupted by noise. In fact, 2 out of 3 of the primitives have to be changed for gesture #1 whereas only 2 out of 5 have to be changed for gesture #2. We therefore normalize the edit distance by dividing the output by the length of the gesture-string, yielding 0.67 for gesture #1 and 0.2 for gesture #2, i.e., gesture #2 is recognized.

The edit distance is a deterministic method but by changing the cost of each of the three operations with respect to likelihoods it becomes a probabilistic method[1]. Concretely we apply the weights described above, see Equation 4.9. These to some extent represent the likelihood of a certain primitive being correct. The higher the weight the more likely a primitive will be. We incorporate the weights into the edit distance method by increasing the score by the weight multiplied by β (a scaling factor) whenever a primitive is *deleted* or *exchanged*. The cost of *inserting* remains 1.

The above principle works for situations where the input sequence only contains one gesture (possibly corrupted by noise). In a real scenario, however, we will

[1]This is related to the Weighted Edit Distance method, which however has fixed weights.

have sequences which are potentially much longer than a gesture and which might contain more gestures after each other. The gesture recognition problem is therefore formulated as for each gesture to find the substring in the detected string, which has the minimum probabilistic edit distance. The recognized gesture will then be the one of the substrings with the minimum distance. Denoting the start point and length of the substring, s and l, respectively, we recognize the gesture present in the detected string as:

$$\text{Gesture} = \arg\min_{k,s,l} PED(\Lambda, k, s, l) \tag{4.10}$$

where k index the different gestures, Λ is the detected string, and $PED(\cdot)$ is the probabilistic edit distance.

4.5 Test and results

For testing purpose we use a vocabulary consisting of 22 primitives. This is illustrated in Figure 4.10. The criteria for finding the primitives are 1) that they represent characteristic and representative 3D configurations, 2) that their configurations contain a certain amount of motion, and 3) that the primitives are used in the description of as many gestures as possible, i.e., fewer primitives are required. By use of this vocabulary of primitives we describe 4 one- and two-arms gestures: "point right", "raise arm", "clap" and "wave".

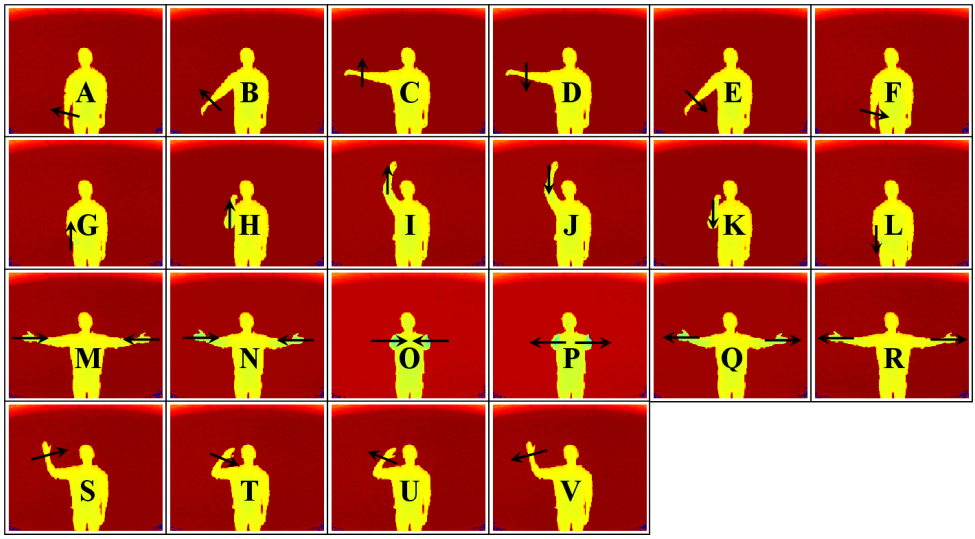


Figure 4.10: The vocabulary consisting of 22 primitives. The primitives are illustrated by range images of the arm configurations and their directions, which are color coded. The color can vary slightly due to error pixels and normalization.

We test the system on data recorded of 10 test subjects, each performing the four gestures 2 times from a 0° and ±45° viewpoint with respect to the camera. A total of 160 video sequences have been recorded. Figure 4.11 shows an example of the visual differences that occur when a gesture is performed from these two viewpoints.

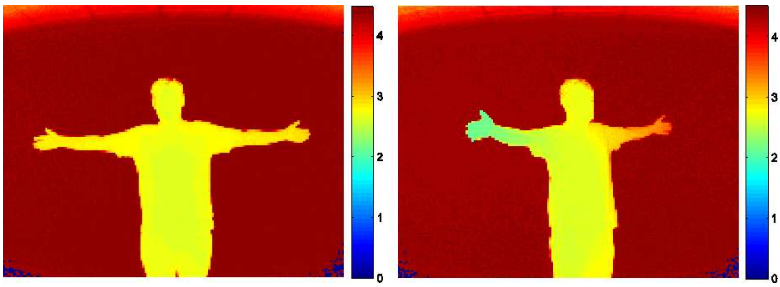


Figure 4.11: Range data examples of a time instance from a video sequence including a person carrying out a "clap" gesture shown from a 0° and +45° camera viewpoint.

To evaluate the view-invariance of the system, the data which is used to train the motion primitives is only from the 0° viewpoint. The overall matching rate is 94.4%. The error distribution can be seen in the confusion matrix in Figure 4.12. In comparison, when only testing on sequences from 0° we obtain a recognition rate of 95.0%.

	1	2	3	4
1. Point right	90.0	5.0	0.0	5.0
2. Raise arm	7.5	92.5	0.0	0.0
3. Clap	0.0	0.0	95.0	5.0
4. Wave	0.0	0.0	0.0	100.0

Figure 4.12: Test results (given in percentages) for the 4 gestures recorded from a 0° and ±45° viewpoint with respect to the camera.

No significant increase in error can be observed when training and testing on sequences from different viewpoints, i.e., the approach supports view-invariant gesture recognition. The errors observed in both tests are mainly due to personal variations when performing gestures like "point right" and "raise arm". I.e., some tend to raise their arm above the shoulder while pointing while some do not stretch their arm fully when raising their arm. Another example is in the case of a "clap" gesture, where one of the arms might not be visible or segmented properly due to a too extreme viewpoint when the individual performs this gesture. Hence a "clap" gesture might be classified to be more likely another gesture. In some of these cases the test person turn more than 45° with respect to the camera. As a result most of

the arm is hidden behind the body and therefore nearly invisible, hence only very little and poorly defined motion is present.

4.5.1 Unknown start and end time

For each sequence we add "noise" in both the beginning and end of the sequence. By doing so, we introduce the realistic problem of having no clear idea about when a gesture commences and terminates which would be the case in a real situation. To achieve a test scenario that resembles this situation we split the gestures into halves and add one of these half gesture to the beginning and one to the end of each gesture to be processed by the system. The added half gestures are chosen randomly resulting in unknown start and end point of the real gesture.

	1	2	3	4
1. Point right	82.5	7.5	2.5	7.5
2. Raise arm	10.0	87.5	0.0	2.5
3. Clap	0.0	2.5	90.0	7.5
4. Wave	2.5	2.5	7.5	87.5

Figure 4.13: Test results when the start and end time for each gesture are unknown (given in percentages) for the 4 gestures recorded from a 0° and ±45° viewpoint with respect to the camera.

Figure 4.13 shows the confusion matrix for the test results with unknown start and end time. The overall recognition rate for this test is 86.9%, and when only testing on sequences from 0° we obtain a recognition rate of 88.8%. The errors are the same as before but with a few additional errors caused by the unknown start and end time of the gestures. Especially, some "wave" gestures seem to cause falsified classifications. The main part of these errors is caused by confusion between "wave" and "clap" gestures performed at ±45°. If the introduced "noise" include the half of a gesture with movements of the arms in front of the body, like a "clap" gesture, this might lead to such errors as the arms have the same start and end positions for these two gestures.

When comparing our ToF-based 3D gesture recognition approach to 2D methods, the main advantages of our approach are that, by applying 3D data, it is able to build a more descriptive representation in comparison to projected 2D data. This also enables view-invariant representation and recognition. In contrast to other view-invariant methods, which rely on multiple calibrated and synchronized cameras followed up by an exhaustive training phase for multiple viewpoints, our approach is able to recognize gestures by using only one ToF sensor (one viewpoint). Furthermore, we are able to handle unknown gesture commencement and termination, along with variation in gesture speed. The 10 test subjects perform gestures at variable execution time, due to how each individual perform a certain gesture. Our

approach is robust in term of gesture speed variation, since the edit distance metric only needs a few correct matches of each primitive, and can handle missing primitive instances, to correctly classify a given gesture. However, it should be noted that more correct primitive matches strengthens the metric due to the assigned probabilities. In comparison to our previous studies [8], the new motion detection carries more information, and the enhanced view-invariant representation (motion context) is more descriptive and distinctive. The results document this by an improvement in the overall recognition rate.

4.6 Conclusion

The contributions of this paper are twofold. Firstly, motion is detected by 2D optical flow estimated in the intensity image but extended to 3D using the depth information acquired from only one viewpoint by a range camera. We show how gestures can be represented efficiently using motion context, and how gesture recognition can be made view-invariant through the use of 3D data and transforming a motion context representation using spherical harmonics. Secondly, for the gesture recognition system we also address the problem of not knowing when a gesture commences and terminates. This is done by recognizing a gesture *not* through a trajectory based approach, but by representing a gesture as a sequence of discrete primitives, and applying a probabilistic edit distance classifier to identify a given gesture.

The presented approach is trained on gestures from 0° viewpoint and tested on gestures seen from both 0° and ±45° viewpoints. The overall recognition rate is 94.4% with known start and end time of gestures, and 86.9% when the start and end time are unknown. These results state the robustness and view-invariance of the system.

A noticeable extension to the current state of this work would be to develop an automatic primitive selection scheme for the training phase. E.g. a clustering technique could be interesting to investigate further for this purpose.

Acknowledgements

This work is partially funded by the MoPrim and the BigBrother projects (Danish National Research Councils - FTP) and partially by the HERMES project (FP6 IST-027110).

References

[1] R. Koch A. Kolb, E. Barth and R. Larsen. Time-of-Flight Sensors in Computer Graphics. In *Eurographics 2009 - State of the Art Reports*, Munich, Germany, March 2009.

[2] M. Ahmad and S.-W. Lee. HMM-based Human Action Recognition using Multiview Image Sequences. In *International Conference on Pattern Recognition*, Hong Kong, August 2006.

[3] H.H. Avils-Arriaga and L.E. Sucar. Dynamic Bayesian Networks for Visual Recognition of Dynamic Gestures. In *Journal of Intelligent and Fuzzy Systems*, 12(3-4):243-250, 2002.

[4] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition using Shape Contexts. In *Pattern Analysis and Machine Intelligence*, 24(4):509-522, 2002.

[5] I. Cohen and H. Li. Inference of Human Postures by Classification of 3D Human Body Shape. In *Workshop on Analysis and Modeling of Faces and Gestures*, Nice, France, October 2003.

[6] R. Larsen D. Hansen and F. Lauze. Improving Face Detection with TOF Cameras. In *International Symposium on Signals, Circuits and Systems*, Iasi, Romania, July 2007.

[7] R. Ronfard D. Weinland and E. Boyer. Free Viewpoint Action Recognition using Motion History Volumes. In *Computer Vision and Image Understanding*, 104(2):249-257, 2006.

[8] M. Holte, T.B. Moeslund, and P. Fihl. Fusion of Range and Intensity Information for View Invariant Gesture Recognition. In *Workshop on Time-of-Flight based Computer Vision*, Anchorage, Alaska, June 2008.

[9] B. Horn and B. Schunck. Determining Optical Flow. In *Artificial Intelligence*, 17:185-203, August 19810.

[10] F. Roca J. Gonzalez, J. Varona and J. Villanueva. aSpaces: Action Spaces for Recognition and Synthesis of Human Actions. In *International Workshop on Articulated Motion and Deformable Objects*, Palma de Mallorca, Spain, November 2002.

[11] O. Jenkins and M. Mataric. Deriving Action and Behavior Primitives from Human Motion Data. In *International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, September 2002.

[12] S. Khan. LUMS School of Science and Engineering Lahore, Pakistan. http://www.cs.ucf.edu/∼khan/.

[13] R. Larsen and B. Lading. Multiple Geodesic Distance Based Registration of Surfaces Applied to Facial Expression Data. In *International Symposium on Signals, Circuits and Systems*, Iasi, Romania, July 2007.

[14] V. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. In *Soviet Physics Doklady*, 10(8):707-710, 1966.

[15] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. In *International Journal of Computer Vision*, 60(2):91-110, November 2004.

[16] B. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of Imaging Understanding Workshop*, Washington DC, USA, April 1981.

[17] T. Martinetz M. Haker, M. Bohme and E. Barth. Geometric Invariants for Facial Feature Tracking with 3D TOF Cameras. In *International Symposium on Signals, Circuits andSystems*, Iasi, Romania, July 2007.

[18] Switzerland. http://www.mesa-imaging.ch MESA Imaging. Technoparkstrasse 1, 8005 Zuerich.

[19] S. Mitra and T. Acharya. Gesture Recognition: A Survey. In *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 37(3):311-324, 2007.

[20] T. Moeslund, A. Hilton, and V. Kruger. A Survey of Advances in Vision-Based Human Motion Capture and Analysis. In *Computer Vision and Image Understanding*, 104(2):90-126, 2006.

[21] E. Rodner O. Kahler and J. Denzler. On Fusion of Range and Intensity Information Using Graph-Cut for Planar Patch Segmentation. In *Dynamic 3D Imaging Workshop*, Heidelberg, Germany, September 2007.

[22] T. Oggier, M. Stamm, M. Schweizer, and J. Pedersen. User Manual SwissRanger 2 Rev. b. Version 1.02, March 2005.

[23] PMD Technologies. Am Eichenhang 50, D-57076 Siegen, Germany. http://www.pmdtec.com.

[24] A. Prusak, I. Schiller, O. Melnychuk, R. Koch, and H. Roth. Pose Estimation and Map Building with a PMD-Camera for Robot Navigation. In *Dynamic 3D Imaging Workshop*, Heidelberg, Germany, September 2007.

[25] C. Pal R. Messing and H. Kautz. Activity Recognition using the Velocity Histories of Tracked Keypoints. In *International Conference on Computer Vision*, Kyoto, Japan, September 2009.

[26] P.K. Reddy, D. Grest, and V. Kruger. Human Action Recognition in Table-top Scenarios: An HMM-based Analysis to Optimize the Performace. In *Computer Analysis of Images and Patterns*, Vienna, Austria, August 2007.

[27] Q. Shi, L. Wang, L. Cheng, and A. Smola. Discriminative Human Action Segmentation and Recognition using Semi-Markov Model. In *Computer Vision and Pattern Recognition*, Anchorage, Alaska, June 2008.

[28] V. Kellokumpu S.N. Vitaladevuni and L.S. Davis. Action Recognition using Ballistic Dynamics. In *Computer Vision and Pattern Recognition*, Anchorage, Alaska, June 2008.

[29] S. Soutschek, J. Penne, J. Hornegger, and J. Kornhuber. 3-D Gesture-Based Scene Navigation in Medical Imaging Applications using Time-of-Flight Cameras. In *Workshop on Time-of-Flight based Computer Vision*, Anchorage, Alaska, June 2008.

[30] R. Souvenir and J. Babbs. Learning the Viewpoint Manifold for Action Recognition. In *Computer Vision and Pattern Recognition*, Anchorage, Alaska, June 2008.

[31] A. Swadsba, B. Liu, J. Penne, O. Jesorsky, and R. Kompe. A Comprehensive System for 3D Modeling from Range Images Acquired from a 3D ToF Sensor. In *International Conference on Computer Vision Systems*, Bielefeld, Germany, March 2007.

[32] A. Swadzba, N. Beuter, J. Schmidt, and G. Sagerer. Tracking Objects in 6D for Reconstructing Static Scenes. In *Workshop on Time-of-Flight based Computer Vision*, Anchorage, Alaska, June 2008.

[33] 3DV Systems. 2nd Carmel St. Industrial Park Building 1, 20692 Yokneam, Israel. http://www.3dvsystems.com.

[34] C. Thurau and V. Hlavac. Pose Primitive Based Human Action Recognition in Videos or Still Images. In *Computer Vision and Pattern Recognition*, Anchorage, Alaska, June 2008.

[35] W.H.A. Wang and C.L. Tung. Dynamic Hand Gesture Recognition using Hierarchical Dynamic Bayesian Networks through Low-level Image Processing. In *International Conference on Machine Learning and Cybernetics*, Kunming, China, July 2008.

[36] M. Minoh Y. Kameda and K. Ikeda. Three Dimensional Motion Estimation of a Human Body Using a Difference Image Sequence. In *Asian Conference on Computer Vision*, Singapore, December 1995.

[37] A. Yilmaz and M. Shah. Actions Sketch: A Novel Action Representation. In *Computer Vision and Pattern Recognition*, San Diego, CA, USA, June 2005.

Chapter 5

Gait Type Analysis

This chapter has previously been published as the chapter "Recognizing Human Gait Types" in the book "Robot Vision" [A]. The chapter presents work on recognition of the three main human gait types, namely walking, jogging, and running. A description of gait types as one unified continuum is also presented. The training data is fully synthetic, meaning that both the motion and the training images are generated by a computer graphics model. Optimizations for the gait type classification method are presented that will allow it to perform in real time and in typical surveillance scenarios.

References

A. P. Fihl and T.B. Moeslund. Recognizing Human Gait Types. In *Robot Vision, ISBN 978953-3070773, In-Tech*, March 2010

Recognizing Human Gait Types

Preben Fihl and Thomas B. Moeslund

Abstract

In this chapter we present a method to describe the continuum of human gait in an invariant manner. The gait description is based on the duty-factor which is adopted from the biomechanics literature. We generate a database of artificial silhouettes representing the three main types of gait, i.e. walking, jogging, and running. By generating silhouettes from different camera angles we make the method invariant to camera viewpoint and to changing directions of movement. Silhouettes are extracted using the Code-book method and represented in a scale- and translation-invariant manner by using shape contexts and tangent orientations. Input silhouettes are matched to the database using the Hungarian method. We define a classifier based on the dissimilarity between the input silhouettes and the gait actions of the database. This classification achieves an overall recognition rate of 87.1% on a diverse test set, which is better than that achieved by other approaches applied to similar data. We extend this classification and results show that our representation of the gait continuum preserves the main features of the duty-factor. The chapter will describe some of the additional challenges that arise when the gait analysis system has to be applied in real scenario. This includes an online parameter adjustment of the background subtration method and estimation of the gait cycle to allow classification on each individual stride. The chapter will also show how gait analysis from multiple cameras can be fused by utilizing tracking of people and how the tracking can enable the system to handle multiple people in the same scene.

5.1 Introduction

Everyday people will observe, analyze, and interpret the motion and actions of the people surrounding them. This is a source of very valuable information about not only what each person is doing but also things like their intentions, their attitude towards the observer, situations they perceive as dangerous or interesting, etc. In open spaces where people are moving around the type of motion will be an important cue for a lot of this information. More specifically, the human gait has been actively investigated in different research areas for this reason.

Within psychology the expression and perception of emotions through styles of motions has been investigated by e.g. [18] and the question of how people infer intention from actions has been studied within neuroscience by e.g. [4]. In biomechanics [2] describes how the choice of different gait types (walking and running) are based on the minimizing of energy consumption in muscles and [31] describes how gait analysis can be used within clinical diagnostics to diagnose a number of diseases.

A lot of the information that is embedded in the gait can be extracted by simply observing a person. Systems that operate around people can benefit greatly from such observations. This fact has driven much research within robotics and computer vision to focus on analysis of human gait with a number of different applications as the aim.

Robots that move and work around humans will be very dependant on their ability to observe people and to interact with humans in an efficient way and the ability to recognize basic human activities is furthermore necessary. Methods for recognizing human gestures to enable natural human-robot interaction has been presented in e.g. [17, 29, 33]. Natural human-robot interaction also requires the robot to behave in a way that is in accordance with the social rules of humans. A method for adapting the robot behavior according to the motion of people is presented in [25]. Since the human gait is a very distinctive type of motion it can be used in many contexts to detect the presence of people, e.g. from surveillance cameras [7, 22, 28]. Gait as a biometric measure has also received much attention because it is non-intrusive [6, 15, 27, 30, 32]. Finally, there has been considerable interest in the computer vision community in the classification of gait types or, more generally, of different types of human action [5, 8, 24]. The research in human action recognition is applicable in a number of areas besides human-robot interaction, e.g. in advanced user interfaces, annotation of video data, intelligent vehicles, and automatic surveillance.

An interesting and challenging area of human gait type recognition is motion in open spaces like town squares, courtyards, or train stations where one of the main human activities is that of gait, i.e. people are walking, jogging, or running. The movements of people in such spaces are however rarely constrained so seen from a camera this will result in challenges like changing direction of motion, significant changes in the scale of people, varying speeds of motion, and often also dynamics backgrounds. This chapter will show how to build a gait type classification system that can handle a number of the challenges that a real-life scenario imposes on such

a gait classification system. i.e. a general system which is *invariant* to camera frame rate and calibration, view point, moving speeds, scale change, and non-linear paths of motion.

Much research concerned with gait attempts to extract features related to the person-specific style of gait whereas this work is concerned with the three general types of gait (walking, jogging and running) and it is therefore more related to the action recognition research than the research on the use of gait in personal identification.

Systems that are invariant to one or more of the factors listed above have been presented in the literature, but so far none has considered all these factors simultaneously. [16] presents good results on classification of different types of human motion but the system is limited to motion parallel to the image plane. [23] describes a method for behavior understanding by combining actions into human behavior. The method handles rather unconstrained scenes but uses the moving speed of people to classify the action being performed. The moving speed cannot be used for gait-type classification. A person jogging along could easily be moving slower than another person walking fast and human observers distinguishing jogging from running do typically not use the speed as a feature. Furthermore, estimation of speed would require scene knowledge that is not always accessible. [5] uses space-time shapes to recognize actions independently of speed. The method is robust to different viewpoints but cannot cope with non-linear paths created by changes in direction of movement. Other state-of-the-art approaches are mentioned in section 5.8 along with a comparison of results.

Current approaches to action classification and gait-type classification consider two or three distinct gait classes, e.g. [16, 23] who consider walking and running, or [5, 8, 24] who consider walking, jogging, and running. However, this distinct classification is not always possible, not even to human observers, and we therefore extend the gait analysis with a more appropriate gait continuum description. Considering gait as a continuum seems intuitive correct for jogging and running, and including walking in such a continuum makes it possible to apply a single descriptor for the whole range of gait types. In this chapter we present a formal description of a gait continuum based on a visual recognizable physical feature instead of e.g. a mixture of probabilities of walking, jogging, and running.

5.1.1 Gait type description based on the Duty-factor

The work presented in this chapter describe the major gait types in a unified gait continuum using the *duty-factor* which is a well established property of gait adopted from the biomechanics literature [2]. To enhance the precision in estimation of the duty-factor we use an effective gait type classifier to reduce the solution space and then calculate the duty-factor within this subspace. The following section will elaborate and motivate our approach.

A current trend in computer vision approaches that deal with analysis of human movement is to use massive amounts of training data, which means spending a lot

of time on extracting and annotating the data and temporally aligning the training sequences. To circumvent these problems an alternative approach can be applied in which computer graphics models are used to generate training data. The advantages of this are very fast training plus the ability to easily generate training data from new viewpoints by changing the camera angle.

In classifying gait types it is not necessary to record a person's exact pose, and silhouettes are therefore sufficient as inputs. Silhouette based methods have been used with success in the area of human identification by gait [6, 15, 30]. The goal in human identification is to extract features that describe the personal variation in gait patterns. The features used are often chosen so that they are invariant to the walking speed and in [32] the same set of features even describe the personal variation in gait patterns of people no matter whether they are walking or running. Inspired by the ability of the silhouette based approaches to describe details in gait, we propose a similar method. Our goal is however quite different from human identification since we want to allow personal variation and describe the different gait types through the duty-factor.

A silhouette based approach does not need a completely realistic looking computer graphics model as long as the shape is correct and the 3D rendering software Poser[1], which has a build-in Walk Designer, can be used to animate human gaits.

To sum up, our approach offers the following three main contributions.

1. The methods applied are chosen and developed to allow for classification in an unconstrained environment. This results in a system that is invariant to more factors than other approaches, i.e. invariant in regard to camera frame rate and calibration, viewpoint, moving speeds, scale change, and non-linear paths of motion.

2. The use of the computer graphics model decouples the training set completely from the test set. Usually methods are tested on data similar to the training set, whereas we train on computer-generated images and test on video data from several different data sets. This is a more challenging task and it makes the system more independent of the type of input data and therefore increases the applicability of the system.

3. The gait continuum is based on a well-established physical property of gait. The duty-factor allows us to describe the whole range of gait types with a single parameter and to extract information that is not dependant on the partially subjective notion of jogging and running.

The remainder of this chapter will first give a thorough introduction of the duty-factor and show its descriptive power. Next, the gait classification framework will be described in detail. The framework is shown in figure 5.1. The human silhouette is

[1]Poser version 6.0.3.140 was used for this work. Currently distributed by Smith Micro Software, Inc.

first extracted (section 5.3) and represented efficiently (section 5.4). We then compare the silhouette with computer graphics silhouettes (section 5.6) from a database (section 5.5). The results of the comparison are calculated for an entire sequence and the gait type and duty-factor of that sequence is extracted (section 5.7). Results are presented in section 5.8 and section 5.9 contains a discussion of these results. Sections 5.10 to 5.12 present solutions to some of the additional challenges that arise when the gait classification system is applied in an online system with multiple cameras, real-time demands, and maintenance of silhouette quality over long time. Section 5.13 concludes the chapter.

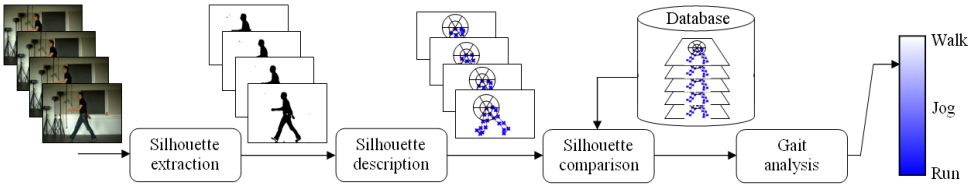


Figure 5.1: An overview of the approach. The main contributions of the method presented here are the computer generated silhouette database, the gait analysis resulting in a gait continuum, and the ability to handle unconstrained environments achieved by the methods applied throughout the system. The gait analysis is further detailed in figure 5.7.

5.2 The Duty-Factor

When a human wants to move fast he/she will run. Running is not simply walking done fast and the different types of gaits are in fact different actions. This is true for vertebrates in general. For example, birds and bats have two distinct flying actions and horses have three different types of gaits. Which action to apply to obtain a certain speed is determined by minimizing some physiological property. For example, turtles seem to optimize with respect to muscle power, horses and humans with respect to oxygen consumption and other animals by minimizing metabolic power. Furthermore, physiological research has shown that the optimal action changes discontinuously with changing speed. [1]

From a computer vision point of view the question is now if *one* (recognizable) descriptor exist, which can represent the continuum of gait. For bipedal locomotion in general, the *duty-factor* can do exactly this. The duty-factor is defined as *"the fraction of the duration of a stride for which each foot remains on the ground"* [2]. Figure 5.2 illustrates the duty-factor in a walk cycle and a run cycle.

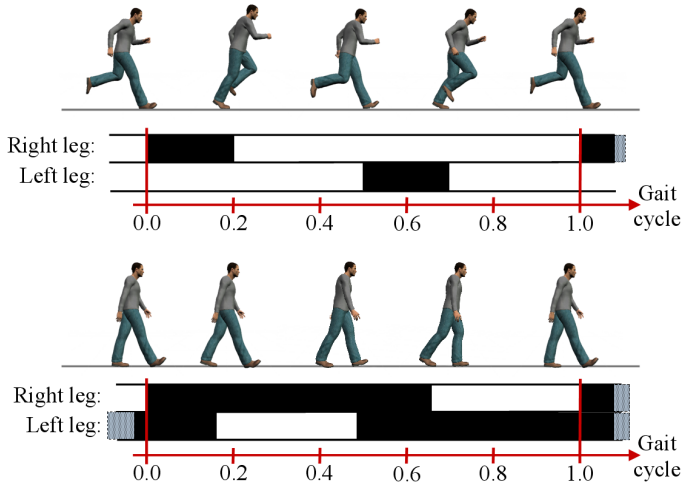


Figure 5.2: Illustration of the duty-factor. The duration of a gait cycle where each foot is on the ground is marked with the black areas. The duty-factor for the depicted run cycle (top) is 0.2 and 0.65 for the depicted walk cycle (bottom).

To illustrate the power of this descriptor we have manually estimated the duty-factor in 138 video sequences containing humans walking, jogging, or running, see figure 5.3. These sequences come from 4 different sources and contain many different individuals entering and exiting at different angles. Some not even following a straight line (see example frames in figure 5.10).

Figure 5.3 shows a very clear separation between walking and jogging/running which is in accordance with the fact that those types of gait are in fact different ways of moving. Jogging and running however, cannot be separated as clearly and there is a gradual transition from one gait type to the other. In fact, the classification of jogging and running is dependent on the observer when considering movements in the transition phase and there exists no clear definition of what separates jogging from running. This problem is apparent in the classification of the sequences used in figure 5.3. Each sequence is either classified by us or comes from a data set where it has been labeled by others. By having more people classify the same sequences it turns out that the classification of some sequences is ambiguous which illustrates the subjectivity in evaluation of jogging and running[2]. [20] reports classification results from 300 video sequences of people walking, jogging, and running. The sequences are classified by several people resulting in classification rates of 100% for walking, 98% for jogging, and only 81% for running, which illustrates the inherent difficulty in distinguishing the two gait types.

[2]The problem of ambiguous classification will be clear when watching for example video sequences from the KTH data set [24], e.g. person 4 jogging in scenario 2 versus person 2 running in scenario 2

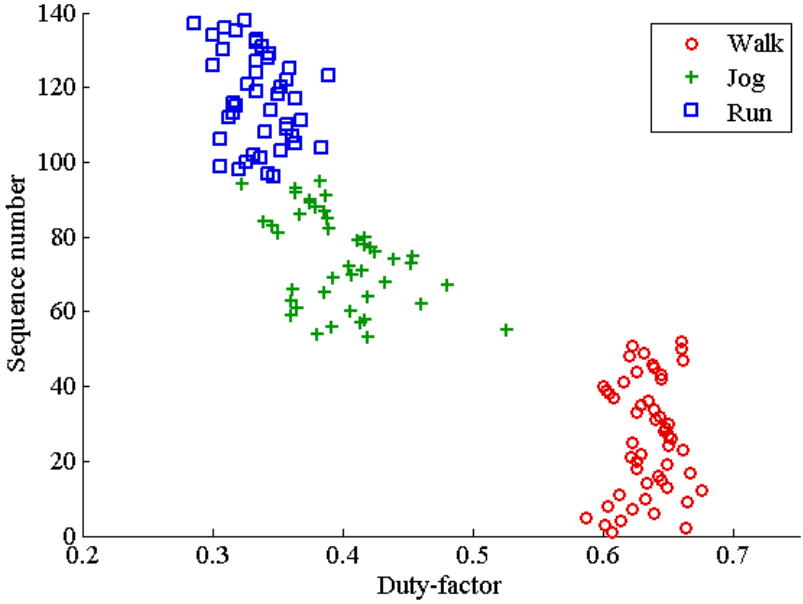


Figure 5.3: The manually annotated duty-factor and gait type for 138 different sequences. Note that the sole purpose of the y-axis is to spread out the data.

With these results in mind we will not attempt to do a traditional classification of walking, jogging, and running which in reality has doubtful ground truth data. Rather, we will use the duty-factor to describe jogging and running as a continuum. This explicitly handles the ambiguity of jogging and running since a video sequence that some people will classify as jogging and other people will classify as running simply map to a point on the continuum described by the duty-factor. This point will not have a precise interpretation in terms of jogging and running but the duty-factor will be precise.

As stated earlier walking and jogging/running are two different ways of moving. However, to get a unified description for all types of gait that are usually performed by people in open spaces we also apply the duty-factor to walking and get a single descriptor for the whole gait continuum.

Even though jogging and running are considered as one gait type in the context of the duty-factor they still have a visual distinction to some extend. This visual distinction is used with some success in the current approaches which classify gait into walking, jogging, and running. We acknowledge the results obtained by this type of approaches and we also propose a new method to classify gait into walking, jogging, and running. In our approach however, this is only an intermediate step to optimize the estimation of the duty-factor which we believe to be the best way of describing gait.

5.3 Silhouette extraction

The first step in the gait analysis framework is to extract silhouettes from the incoming video sequences. For this purpose we do foreground segmentation using the Codebook background subtraction method as described in [9] and [10]. This method has been shown to be robust in handling both foreground camouflage and shadows. This is achieved by separating intensity and chromaticity in the background model. Moreover, the background model is multi modal and multi layered which allows it to model moving backgrounds such as tree branches and objects that become part of the background after staying stationary for a period of time. To maintain good background subtraction quality over time it is essential to update the background model and [9] describes two different update mechanisms to handle rapid and gradual changes respectively. By using this robust background subtraction method we can use a diverse set of input sequences from both indoor and outdoor scenes.

5.4 Silhouette description

When a person is moving around in an unconstrained scene his or her arms will not necessarily swing in a typical "gait" manner; the person may be making other gestures, such as waving, or he/she might be carrying an object. To circumvent the variability and complexity of such scenarios we choose to classify the gait solely on the silhouette of the legs. Furthermore, [14] shows that identification of people on the basis of gait, using the silhouette of legs alone, works just as well as identification based on the silhouette of the entire body.

To extract the silhouette of the legs we find the height of the silhouette of the entire person and use the bottom 50% as the leg silhouette. Without loss of generality this approach avoids errors from the swinging hands below the hips, although it may not be strictly correct from an anatomic point of view. To reduce noise along the contour we apply morphological operations to the silhouette. Some leg configurations cause holes in the silhouette, for example running seen from a non-side view in figure 5.5(c). Such holes are descriptive for the silhouette and we include the contour of these holes in the silhouette description.

To allow recognition of gait types across different scales we use shape contexts and tangent orientations [3] to describe the leg silhouettes. n points are sampled from the contour of the leg silhouette and for each point we determine the shape context and the tangent orientation at that point, see figure 5.4. With K bins in the log-polar histogram of the shape context we get an $n\times(K+1)$ matrix describing each silhouette. Scale invariance is achieved with shape contexts by normalizing the size of the histograms according to the mean distance between all point pairs on the contour. Specifically, the normalizing constant q used for the radial distances of the

histograms are defined as follows:

$$q = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} |p_i - p_j| \tag{5.1}$$

where n is the number of points p sampled from the contour.

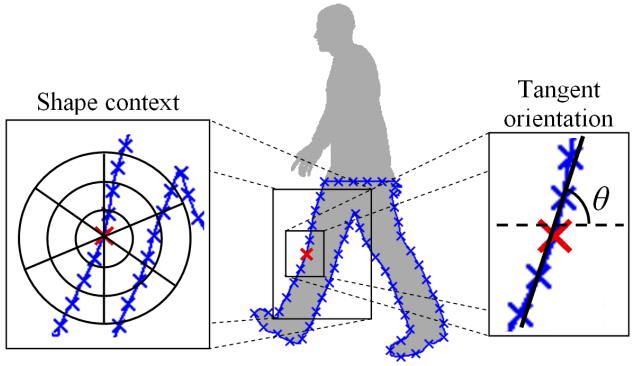


Figure 5.4: Illustration of the silhouette description. The crosses illustrate the points sampled from the silhouette. Shape contexts and tangent orientations are used to describe the silhouette.

5.5 Silhouette database

To represent our training data we create a database of human silhouettes performing one cycle of each of the main gait types: walking, jogging, and running. To make our method invariant to changes in viewpoint we generate database silhouettes from three different camera angles. With 3D-rendering software this is an easy and very rapid process that does not require us to capture new real-life data for statistical analysis. The database contains silhouettes of the human model seen from a side view and from cameras rotated 30 degrees to both sides. The combination of the robust silhouette description and three camera angles enable the method to handle diverse moving directions and oblique viewing angles. Specifically, database silhouettes can be matched with silhouettes of people moving at angles of at least ±45 degrees with respect to the viewing direction. People moving around in open spaces will often change direction while in the camera's field of view (creating non-linear paths of motion), thus we cannot make assumptions about the direction of movement. To handle this variability each new input silhouette is matched to database silhouettes taken from all camera angles. Figure 5.10, row 1 shows a sequence with a non-linear motion path where the first frame will match database silhouettes from a viewpoint of -30 degrees and the last frame will match database silhouettes from a viewpoint

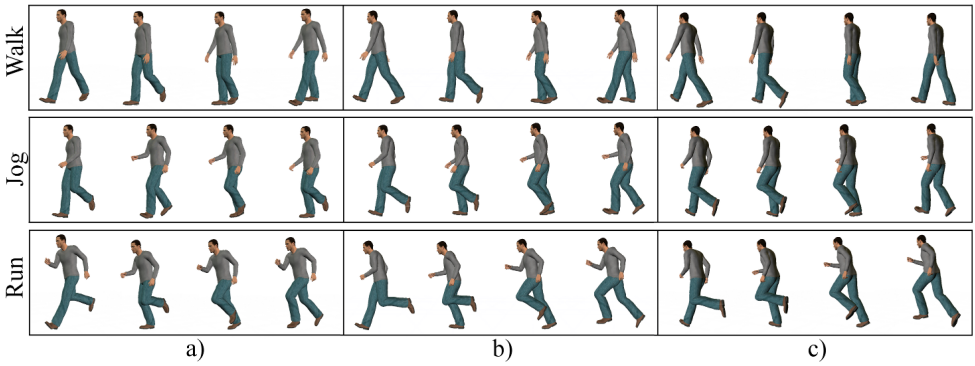


Figure 5.5: Example of database silhouettes generated by 3D-rendering software. Silhouettes are generated from three viewpoints. a) and c) illustrate renderings from cameras rotated 30 degrees to each side. b) illustrates renderings from a direct side view.

of 30 degrees. The silhouettes generated are represented as described in section 5.4. We generate T silhouettes of a gait cycle for each of the three gait types. This is repeated for the three viewpoints, i.e. $T \cdot 3 \cdot 3$ silhouettes in total. Figure 5.5 shows examples of the generated silhouettes.

Each silhouette in the database is annotated with the number of feet in contact with the ground which is the basis of the duty-factor calculation.

To analyze the content of the database with respect to the ability to describe gait we created an Isomap embedding [26] of the shape context description of the silhouettes. Based on the cyclic nature of gait and the great resemblance between gait types we expect that gait information can be described by some low dimensional manifold. Figure 5.6 shows the 2-dimensional embedding of our database with silhouettes described by shape contexts and tangent orientations and using the costs resulting from the Hungarian method (described in section 5.6) as distances between silhouettes.

According to figure 5.6 we can conclude that the first two intrinsic parameters of the database represent 1) the total distance between both feet and the ground and 2) the horizontal distance between the feet. This reasonable 2-dimensional representation of the database silhouettes shows that our description of the silhouettes and our silhouette comparison metric does capture the underlying manifold of gait silhouettes in a precise manner. Hence, gait type analysis based on our silhouette description and comparison seems promising.

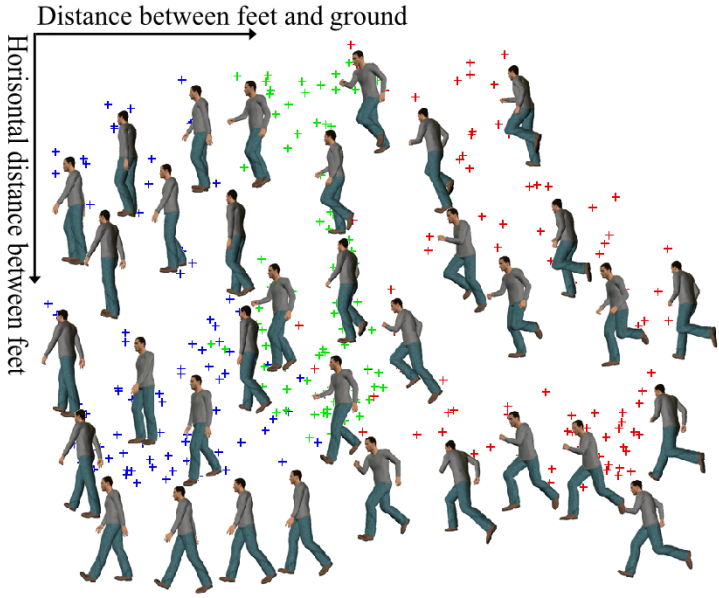


Figure 5.6: Illustration of the ISOMAP embedding and a representative subset of the database silhouettes.

5.6 Silhouette comparison

To find the best match between an input silhouette and database silhouettes we follow the method of [3]. We calculate the cost of matching a sampled point on the input silhouette with a sampled point on a database silhouette using the χ^2 test statistics. The cost of matching the shape contexts of point p_i on one silhouette and point p_j on the other silhouette is denoted $c_{i,j}$. The normalized shape contexts at points p_i and p_j are denoted $h_i(k)$ and $h_j(k)$ respectively with k as the bin number, $k = 1, 2, ..., K$. The χ^2 test statistics is given as:

$$c_{i,j} = \frac{1}{2} \sum_{k=1}^{K} \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)} \tag{5.2}$$

The normalized shape contexts gives $c_{i,j} \in [0; 1]$.

The difference in tangent orientation $\phi_{i,j}$ between points p_i and p_j is normalized and added to $c_{i,j}$ ($\phi_{i,j} \in [0; 1]$). This gives the final cost $C_{i,j}$ of matching the two points:

$$C_{i,j} = a \cdot c_{i,j} + b \cdot \phi_{i,j} \tag{5.3}$$

where a and b are weights. Experiments have shown that $\phi_{i,j}$ effectively discriminates points that are quite dissimilar whereas $c_{i,j}$ expresses more detailed differences which should have a high impact on the final cost only when tangent orientations are

alike. According to this observation we weight the difference in tangent orientation $\phi_{i,j}$ higher than shape context distances $c_{i,j}$. Preliminary experiments show that the method is not too sensitive to the choice of these weights but a ratio of 1 to 3 yields good results, i.e. $a = 1$ and $b = 3$

The costs of matching all point pairs between the two silhouettes are calculated. The Hungarian method [19] is used to solve the square assignment problem of identifying which one-to-one mapping between the two point sets that minimizes the total cost. All point pairs are included in the cost minimization, i.e. the ordering of the points is not considered. This is because points sampled from a silhouette with holes will have a very different ordering compared to points sampled from a silhouette without holes but with similar leg configuration, see row three of figure 5.5(c) (second and third image) for an example.

By finding the best one-to-one mapping between the input silhouette and each of the database silhouettes we can now identify the best match in the whole database as the database silhouette involving the lowest total cost.

5.7 Gait analysis

The gait analysis consists of two steps. First we do classification into one of the three gait types, i.e. walking, jogging, or running. Next we calculate the duty-factor D based on the silhouettes from the classified gait type. This is done to maximize the likelihood of a correct duty-factor estimation. Figure 5.7 illustrates the steps involved in the gait type analysis. Note that the silhouette extraction, silhouette description, and silhouette comparison all process a single input frame at a time whereas the gait analysis is based on a sequence of input frames.

To get a robust classification of the gait type in the first step we combine three

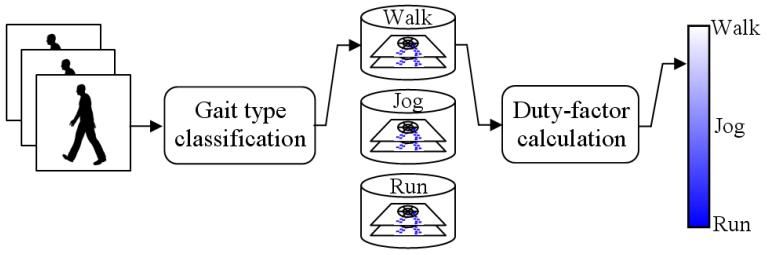


Figure 5.7: An overview of the gait analysis. The figure shows the details of the block "Gait analysis" in figure 5.1. The output of the silhouette comparison is a set of database silhouettes matched to the input sequence. In the gait type classification these database silhouettes are classified as a gait type which defines a part of the database to be used for the duty-factor calculation.

different types of information. We calculate an *action error* E for each action and two associated weights: *action likelihood* α and *temporal consistency* β. The following subsections describe the gait analysis in detail starting with the action error and the two associated weights followed by the duty-factor calculation.

5.7.1 Action error

The output of the silhouette comparison is a set of distances between the input silhouette and each of the database silhouettes. These distances express the difference or error between two silhouettes. Figure 5.8 illustrates the output of the silhouette comparison. The database silhouettes are divided into three groups corresponding to walking, jogging, and running, respectively. We accumulate the errors of the best matches within each group of database silhouettes. These accumulated errors constitute the *action error* E and corresponds to the difference between the action being performed in the input video and each of the three actions in the database, see figure 5.9.

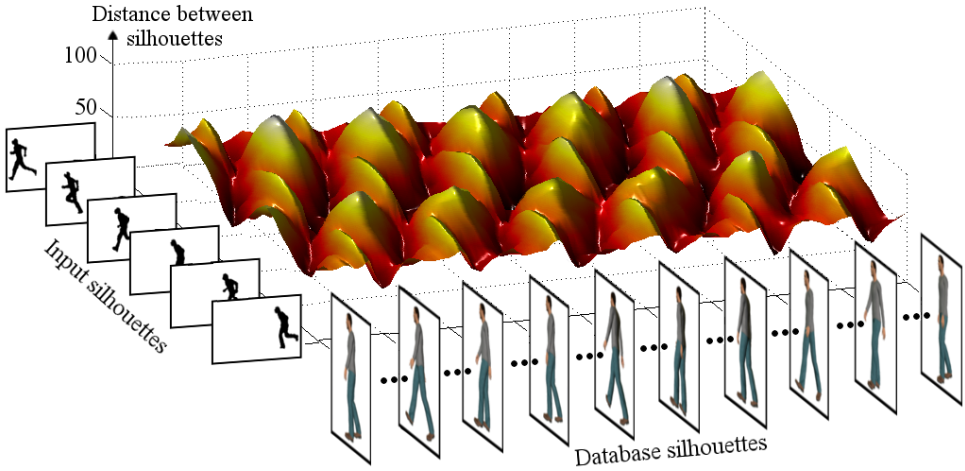


Figure 5.8: Illustration of the silhouette comparison output. The distances between each input silhouette and the database silhouettes of each gait type are found (shown for walking only). 90 database silhouettes are used per gait type, i.e. $T = 30$.

5.7.2 Action likelihood

When silhouettes of people are extracted in difficult scenarios and at low resolutions the silhouettes can be noisy. This may result in large errors between the input

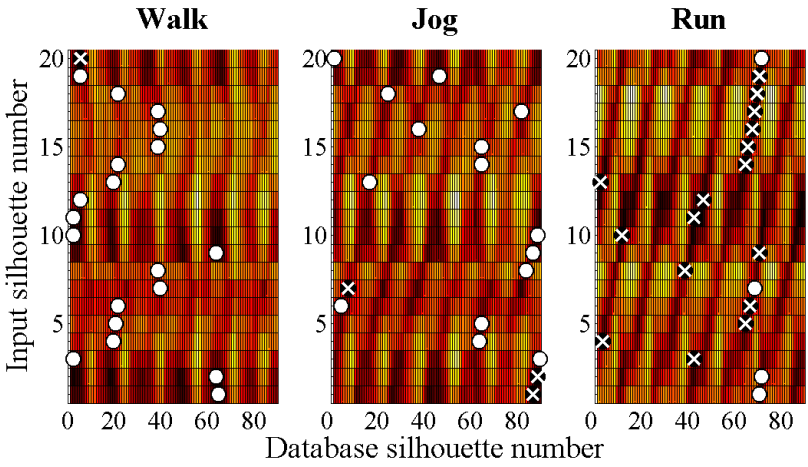


Figure 5.9: The output of the silhouette comparison of figure 5.8 is shown in 2D for all gait types (dark colors illustrate small errors and bright colors illustrate large errors). For each input silhouette the best match among silhouettes of the same action is marked with a white dot and the best overall match is marked with a white cross. The shown example should be interpreted as follows: the silhouette in the first input frame is closest to walking silhouette number 64, to jogging silhouette number 86, and to running silhouette number 70. These distances are used when calculating the action error. When all database silhouettes are considered together, the first input silhouette is closest to jogging silhouette number 86. This is used in the calculation of the two weights.

silhouette and a database silhouette, even though the actual pose of the person is very similar to that of the database silhouette. At the same time, small errors may be found between noisy input silhouettes and database silhouettes with quite different body configurations (somewhat random matches). To minimize the effect of the latter inaccuracies we weight the action error by the likelihood of that action. The action likelihood of action a is given as the percentage of input silhouettes that match action a better than the other actions. Since we use the minimum action error the actual weight applied is one minus the action likelihood:

$$\alpha_a = 1 - \frac{n_a}{N} \tag{5.4}$$

where n_a is the number of input silhouettes in a sequence with the best overall match to a silhouette from action a, and N is the total number of input silhouettes in that video sequence. This weight will penalize actions that have only a few overall best matches, but with small errors, and will benefit actions that have many overall best matches, e.g. the running action in figure 5.9.

5.7.3 Temporal consistency

When considering only the overall best matches we can find sub-sequences of the input video where all the best matches are of the same action *and* in the right order with respect to a gait cycle. This is illustrated in figure 5.9 where the running action has great temporal consistency (silhouette numbers 14-19). The database silhouettes are ordered in accordance with a gait cycle. Hence, the straight line between the overall best matches for input silhouettes 14 to 19 shows that each new input silhouette matches the database silhouette that corresponds to the next body configuration of the running gait cycle.

Sub-sequences with correct temporal ordering of the overall best matches increase our confidence that the action identified is the true action. The temporal consistency describes the length of these sub-sequences. Again, since we use the minimum action error we apply one minus the temporal consistency as the weight β_a:

$$\beta_a = 1 - \frac{m_a}{N} \tag{5.5}$$

where m_a is the number of input silhouettes in a sequence in which the best overall match has correct temporal ordering within action a, and N is the total number of input silhouettes in that video sequence.

Our definition of temporal consistency is rather strict when you consider the great variation in input silhouettes caused by the unconstrained nature of the input. A strict definition of temporal consistency allows us to weight it more highly than action likelihood, i.e. we apply a scaling factor w to β to increase the importance of temporal consistency in relation to action likelihood:

$$\beta_a = 1 - w \cdot \frac{m_a}{N} \tag{5.6}$$

5.7.4 Gait-type classification

The final classifier for the gait type utilizes both the action likelihood and the temporal consistency as weights on the action error. This yields:

$$Action = \arg\min_a (E_a \cdot \alpha_a \cdot \beta_a) \tag{5.7}$$

where E_a is the action error, α_a is the action likelihood, β_a is the weighted temporal consistency.

5.7.5 Duty-Factor calculation

As stated earlier the duty-factor is defined as the fraction of the duration of a stride for which each foot remains on the ground. Following this definition we need to identify the duration of a stride and for how long each foot is in contact with the ground.

A stride is defined as one complete gait cycle and consists of two steps. A stride can be identified as the motion from a left foot takeoff (the foot leaves the ground) and until the next left foot takeoff (see figure 5.2 for an illustration). Accordingly a step can be identified as the motion from a left foot takeoff to the next right foot takeoff. Given this definition of a step it is natural to identify steps in the video sequence by use of the silhouette width. From a side view the silhouette width of a walking person will oscillate in a periodic manner with peaks corresponding to silhouettes with the feet furthest apart. The interval between two peaks will (to a close approximation) define one step [6]. This also holds for jogging and running and can furthermore be applied to situations with people moving diagonally with respect to the viewing direction. By extracting the silhouette width from each frame of a video sequence we can identify each step (peaks in silhouette width) and hence determine the mean duration of a stride t_s in that sequence.

For how long each foot remains on the ground can be estimated by looking at the database silhouettes that have been matched to a sequence. We do not attempt to estimate ground contact directly in the input videos which would require assumptions about the ground plane and camera calibrations. For a system intended to work in unconstrained open scenes such requirements will be a limitation to the system. In stead of estimating the feet's ground contact in the input sequence we infer the ground contact from the database silhouettes that are matched to that sequence. Since each database silhouette is annotated with the number of feet supported on the ground this is a simple lookup in the database. The ground support estimation is based solely on silhouettes from the gait type found in the gait-type classification which maximize the likelihood of a correct estimate of the ground support.

The total ground support G of both feet for a video sequence is the sum of ground support of all the matched database silhouettes within the specific gait type. To get the ground support for each foot we assume a normal moving pattern (not limping, dragging one leg, etc.) so the left and right foot have equal ground support and the mean ground support g for each foot during one stride is $\frac{G}{2 \cdot n_s}$, where n_s is the number of strides in the sequence. The duty-factor D is now given as $D = \frac{g}{t_s}$. In summary we have

$$\text{Duty-factor} \quad D = \frac{G}{2 \cdot n_s \cdot t_s} \tag{5.8}$$

where G is the total ground support, n_s is the number of strides, and t_s is the mean duration of a stride in the sequence.

The manual labeled data of figure 5.3 allows us to further enhance the precision of the duty-factor description. It can be seen from figure 5.3 that the duty-factor for running is in the interval [0.28;0.39] and jogging is in the interval [0.34;0.53]. This can not be guarantied to be true for all possible executions of running and jogging but the great diversity in the manually labeled data allows us to use these intervals in the duty-factor estimation. Since walking clearly separates from jogging and running and since no lower limit is needed for running we infer the following

constraints on the duty factor of running and jogging:

$$\begin{aligned} D_{running} &\in [0; 0.39] \\ D_{jogging} &\in [0.34; 0.53] \end{aligned}$$

We apply these bounds as a post-processing step. If the duty-factor of a sequence lies outside one of the appropriate bounds then the duty-factor will be assigned the value of the exceeded bound.

5.8 Results

To emphasize the contributions of our two-step gait analysis we present results on both steps individually and on the gait continuum achieved by combining the two steps.

A number of recent papers have reported good results on the classification of gait types (often in the context of human action classification). To compare our method to these results and to show that the gait type classification is a solid base for the duty-factor calculation we have tested this first step of the gait analysis on its own. After this comparison we test the duty-factor description with respect to the ground truth data shown in figure 5.3, both on its own and in combination with the gait type classification.

The tests are conducted on a large and diverse data set. We have compiled 138 video sequences from 4 different data sets. The data sets cover indoor and outdoor video, different moving directions with respect to the camera (up to ± 45 degrees from the viewing direction), non-linear paths, different camera elevations and tilt angles, different video resolutions, and varying silhouette heights (from 41 pixels to 454 pixels). Figure 5.10 shows example frames from the input videos. Ground truth gait types were adopted from the data sets when available and manually assigned by us otherwise.

For the silhouette description the number of sampled points n was 100 and the number of bins in the shape contexts K was 60. 30 silhouettes were used for each gait cycle, i.e., $T = 30$. The temporal consistency was weighted by a factor of four determined through quantitative experiments, i.e. $w = 4$.

5.8.1 Gait-type classification

When testing only the first step of the gait analysis we achieve an overall recognition rate of 87.1%. Table 5.1 shows the classification results in a confusion matrix.

The matching percentages in table 5.1 cannot directly be compared to the results of others since we have included samples from different data sets to obtain more

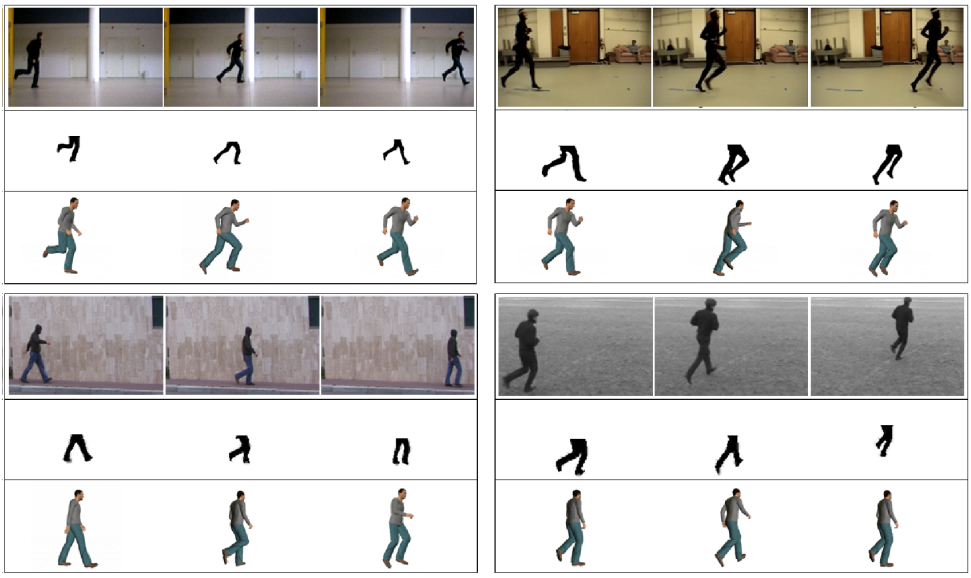


Figure 5.10: Samples from the 4 different data sets used in the test together with the extracted silhouettes of the legs used in the database comparison, and the best matching silhouette from the database. Top left: data from our own data set. Bottom left: data from the Weizmann data set [5]. Top right: data from the CMU data set obtained from mocap.cs.cmu.edu. The CMU database was created with funding from NSF EIA-0196217. Bottom right: data from the KTH data set [24].

diversity. However, 87 of the sequences originate from the KTH data set [24] and a loose comparison is possible on this subset of our test sequences. In table 5.2 we list the matching results of different methods working on the KTH data set. The KTH data set remains one of the largest data sets of human actions in terms of number of test subjects, repetitions, and scenarios and many papers have been published with results on this data set, especially within the last two years. A number of different test setups have been used which makes a direct comparison impossible and we therefore merely list a few of the best results to show the general level of recognition rates. We acknowledge that the KTH data set contains three additional

	Walk	Jog	Run
Walk	96.2	3.8	0.0
Jog	0.0	65.9	34.1
Run	0.0	2.6	97.4

Table 5.1: Confusion matrix for the gait type classification results.

Methods	Classification results in %			
	Total	Walk	Jog	Run
[11]*	92.3	99	90	88
Our method	92.0	100.0	80.6	96.3
[13]*	89.0	88	89	90
[12]*	89.3	99	89	80
[20]	84.3	98	79	76
[24]*	75.0	83.8	60.4	54.9

Table 5.2: Best reported classification results on the KTH data set. The matching results of our method are based on the 87 KTH sequences included in our test set. * indicate that the method work on all actions of the KTH data set.

actions (boxing, hand waving, and hand clapping) and that some of the listed results include these. However, for the results reported in the literature the gait actions are in general not confused with the three hand actions. The results can therefore be taken as indicators of the ability of the methods to classify gait actions exclusively.

Another part of our test set is taken from the Weizmann data set [5]. They classify nine different human actions including walking and running but not jogging. They achieve a near perfect recognition rate for running and walking and others also report 100% correct recognitions on this data set, e.g. [21]. To compare our results to this we remove the jogging silhouettes from the database and leave out the jogging sequences from the test set. In this walking/running classification we achieve an overall recognition rate of 98.9% which is slightly lower. Note however that the data sets we are testing on include sequences with varying moving directions where the results in [5] and [21] are based on side view sequences.

In summary, the recognition results of our gait-type classification provides a very good basis for the estimation of the duty-factor.

5.8.2 Duty-factor

To test our duty-factor description we estimate it automatically in the test sequences. To show the effect of our combined gait analysis we first present results for the duty-factor estimated without the preceding gait-type classification to allow for a direct comparison.

Figure 5.11 shows the resulting duty-factors when the gait type classification is not used to limit the database silhouettes to just one gait type. Figure 5.12 shows the estimated duty-factors with our two-step gait analysis scheme. The estimate of the duty-factor is significantly improved by utilizing the classification results of the gait type classification. The mean error for the estimate is 0.050 with a standard deviation of 0.045.

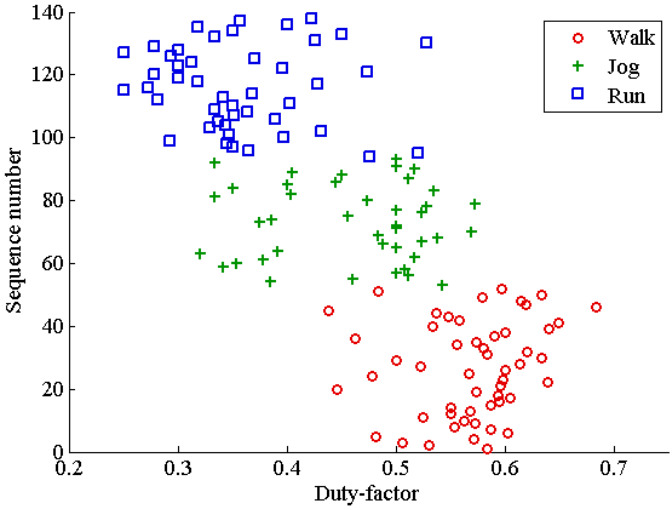


Figure 5.11: The automatically estimated duty-factor from the 138 test sequences without the use of the gait type classification. The y-axis solely spreads out the data.

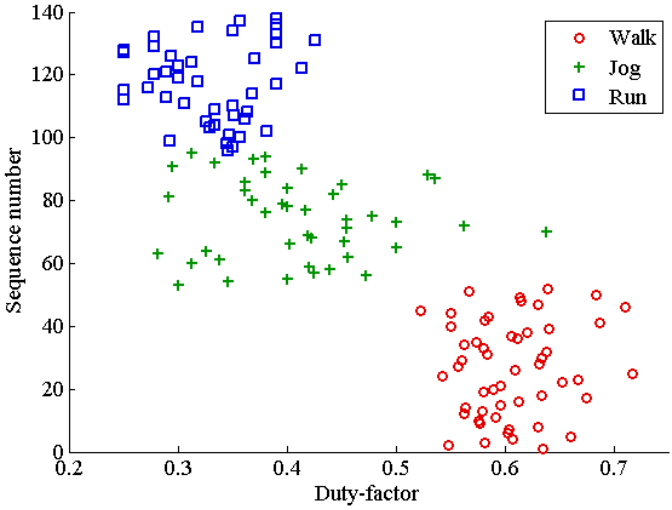


Figure 5.12: The automatically estimated duty-factor from the 138 test sequences when the gait type classification has been used to limit the database to just one gait type. The y-axis solely spreads out the data.

5.9 Discussion

When comparing the results of the estimated duty-factor (figure 5.12) with the ground truth data (figure 5.3) it is clear that the overall tendency of the duty-factor is reproduced with the automatic estimation. The estimated duty-factor has greater variability mainly due to small inaccuracies in the silhouette matching. A precise estimate of the duty-factor requires a precise detection of when the foot actually touches the ground. However, this detection is difficult because silhouettes of the human model are quite similar just before and after the foot touches the ground. Inaccuracies in the segmentation of the silhouettes in the input video can make for additional ambiguity in the matching.

The difficulty in estimating the precise moment of ground contact leads to considerations on alternative measures of a gait continuum, e.g. the Froude number [1] that is based on walking speed and the length of the legs. However, such measures requires information about camera calibration and the ground plane which is not always accessible with video from unconstrained environments. The processing steps involved in our system and the silhouette database all contributes to the overall goal of creating a system that is invariant to usual challenges in video from unconstrained scenes and a system that can be applied in diverse setups without requiring additional calibrations.

The misclassifications of the three-class classifier also affect the accuracy of the estimated duty-factor. The duty-factor of the four jogging sequences misclassified as walking disrupt the perfect separation of walking and jogging/running expected from the manually annotated data. All correctly classified sequences however maintain this perfect separation.

To test wether the presented gait classification framework provides the kind of invariance that is required for unconstrained scenes we have analyzed the classification errors in table 5.1. This analysis shows no significant correlation between the classification errors and the camera viewpoint (pan and tilt), the size and quality of the silhouettes extracted, the image resolution, the linearity of the path, and the amount of scale change. Furthermore, we also evaluated the effect of the number of frames (number of gait cycles) in the sequences and found that our method classifies gait types correctly even when there are only a few cycles in the sequence. This analysis is detailed in table 5.3 which shows the result of looking at a subset of the test sequences containing a specific video characteristic.

A number of the sequences in table 5.3 have more than one of the listed characteristics (e.g. small silhouettes in low resolution images) so the error percentages are somewhat correlated. It should also be noted that the gait type classification results in only 17 errors which gives a relatively small number of sequences for this analysis. However, the number of errors in each subset corresponds directly to the number of sequences in that subset which is a strong indication that our method is indeed invariant to the main factors relevant for gait classification.

Video characteristic		Percentage of sequences	Percentage of errors
Non-side view		43	41
Small silhouettes	(1)	58	59
Low resolution images	(2)	63	65
Non linear path		3	0
Significant scale change	(3)	41	41
Less than 2 strides		43	41

Table 5.3: The table shows how different video characteristics effect the classification errors, e.g. 43% of the sequences have a non-side view and these sequences account for 41% of the errors. The results are based on 138 test sequences out of which 17 sequences were erroneously classified. Notes: (1): Mean silhouette height of less than 90 pixels. (2): Image resolution of 160x120 or smaller. (3): Scale change larger than 20% of the mean silhouette height during the sequence.

The majority of the errors in table 5.1 occur simply because the gait type of jogging resembles that of running which supports the need for a gait continuum.

5.10 Multi camera setup

The system has been designed to be invariant towards the major challenges in a realistic real-world setup. Regarding invariance to view point, we have achieved this for gait classification of people moving at an angle of up to ±45 degrees with respect to the view direction. The single-view system can however easily be extended to a multi-view system with synchronized cameras which can allow for gait classification of people moving at completely arbitrary directions. A multi-view system must analyze the gait based on each stride rather than a complete video sequence since people may change both moving direction and type of gait during a sequence.

The direction of movement can be determined in each view by tracking the people and analyzing the tracking data. Tracking is done as described in [9]. If the direction of movement is outside the ±45 degree interval then that view can be excluded. The duration of a stride can be determined as described in section 5.7.5 from the view where the moving direction is closest to a direct side-view. The gait classification results of the remaining views can be combined into a multi-view classification system by extending equations 5.7 and 5.8 into the following and doing the calculations based on the last stride in stead of the whole sequence.

$$Action = \arg\min_{a}(\sum_{V} E_a \cdot \alpha_a \cdot \beta_a) \tag{5.9}$$

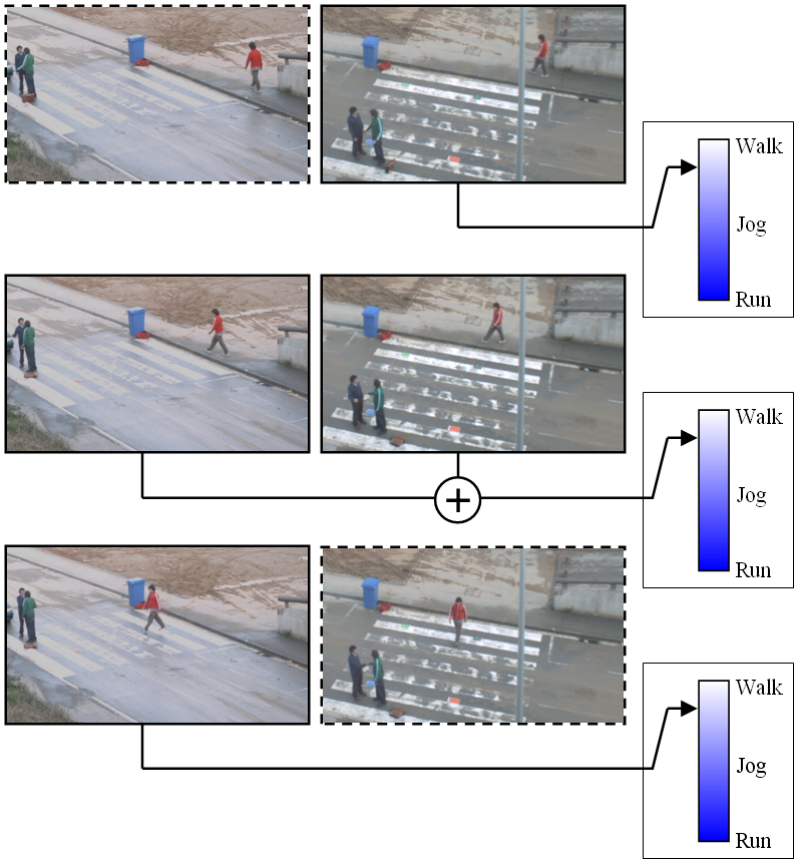


Figure 5.13: A two-camera setup. The figure shows three sets of synchronized frames from two cameras. The multi-camera gait classification enables the system to do classification based on either one view (top and bottom frames) or a combination of both views (middle frame).

$$D = \frac{1}{n_V} \cdot \sum_{V} D_v \tag{5.10}$$

where V is the collection of views with acceptable moving directions, E_a is the action error, α_a is the action likelihood, β_a is the temporal consistency, D is the duty-factor, n_V is the number of views, and D_v is the duty-factor from view v.

Figure 5.13 illustrates a two-camera setup where the gait classification is based on either one of the cameras or a combination of both cameras.

5.11 Real time performance

The full potential of the gait analysis framework can only be achieved with real-time performance. Non-real-time processing can be applied for annotation of video data but for e.g. human-robot interaction, automated video surveillance, and intelligent vehicles real-time performance is necessary.

Real-time performance can be achieved with an optimized implementation and minor changes in the method. The extraction of the contour of the silhouettes is limited to the outermost contour. Disregarding the inner contours (see figure 5.14) gave a decrease in processing time but also a small decrease in classification results due to the loss of details in some silhouettes.

The most time consuming task of the gait classification is the matching of the input silhouette to the database silhouettes both represented in terms of Shape Contexts. By decreasing the number of points sampled around the contour from 100 points to 20 points and by decreasing the number of bins in the Shape Contexts from 60 to 40 the processing time is significantly improved while still maintaining most of the descriptive power of the method.

With these changes the gait classification system is running at 12-15 frames per second on a standard desktop computer with a 2GHz dual core processor and 2GB of RAM. This however also means a decrease in the classification power of the system. When looking at the gait type classification a recognition rate of 83.3% is achieved with the real-time setup compared to 87.1% with the original setup. The precision of the duty-factor estimation also decreases slightly. This decrease in recognition rate is considered to be acceptable compared to the increased applicability of a real-time system.

Figure 5.14: Left: the input silhouette. Middle: the outermost contour extracted in the real time system. Right: the contour extracted in the original system.

5.12 Online parameter tuning of segmentation

The silhouette extraction based on the Codebook background subtraction is a critical component in the system. Noise in the extracted silhouettes has a direct impact on the classification results. Illumination and weather conditions can change rapidly

in unconstrained open spaces so to ensure the performance of the background subtraction in a system receiving live input directly from a camera we have developed a method for online tuning of the segmentation.

The performance of the Codebook background subtraction method is essentially controlled by three parameters; two controlling the allowed variation in illumination and one controlling the allowed variation in chromaticity. The method is designed to handle shadows so with a reasonable parameter setup the Codebook method will accept relatively large variations in illumination to account for shadows that are cast on the background. However, changes in lighting conditions in outdoor scenes also have an effect on the chromaticity level which is not directly modeled in the method. Because of this, the parameter that controls the allowed variation in chromaticity σ is the most important parameter to adjust online (i.e. fixed parameters for the illumination variation will handle changing lighting conditions well, whereas a fixed parameter for the chromaticity variation will not).

To find the optimal setting for σ at runtime we define a quality measure to evaluate a specific value of σ and by testing a small set of relevant values for each input frame we adjust σ by optimizing this quality measure.

The quality measure is based on the difference between the edges of the segmentation and the edges of the input image. An edge background model is acquired simultaneously with the Codebook background model which allows the system to classify detected edges in a new input frame as either foreground or background edges. The map of foreground edges has too much noise to be used for segmentation itself but works well when used to compare the quality of different foreground segmentations of the same frame. The quality score Q is defined as follows:

$$Q = \frac{\sum E_{fg} \cdot E_{seg}}{\sum E_{seg}} \tag{5.11}$$

where E_{fg} are the foreground edges and E_{seg} are the edges of the foreground mask from the background subtraction. So the quality score describes the fraction of edges from the foreground mask that corresponds to foreground edges from the input image.

The background subtraction is repeated a number of times on each input frame with varying values of σ and the quality score is calculated after each repetition. The segmentation that results in the highest quality score is used as the final segmentation. Figures 5.15 and 5.16 show example images of this process.

The repetitive segmentation of each frame slows the silhouette extraction of the gait classification system down but by only testing a few values of σ for each frame real time performance can still be achieved. The first frames of a new input sequence will be tested with up to 30 values of σ covering a large interval (typically [1:30]) to initialize the segmentation whereas later frames will be tested with only four to six values of σ in the range ± 2 of the σ-value from the previous frame.

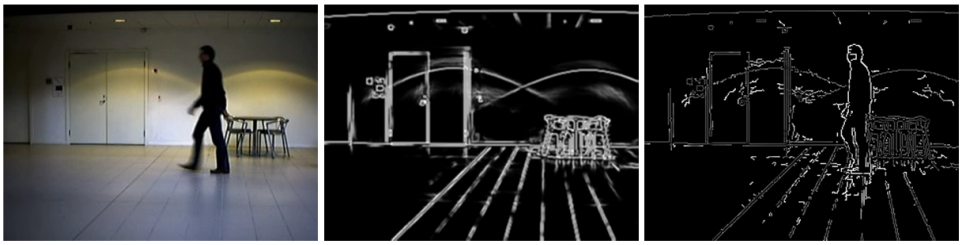


Figure 5.15: Left: the input image. Middle: the background edge model. Right: the foreground edges.

Figure 5.16: Three segmentation results with varying values of σ. Left: σ-value too low. Middle: optimal σ-value. Right: σ-value too high.

5.13 Conclusion

The gait type of people that move around in open spaces is an important property to recognize in a number of applications, e.g. automated video surveillance and human-robot interaction. The classical description of gait as three distinct types is not always adequate and this chapter has presented a method for describing gait types with a gait continuum which effectively extends and unites the notion of running, jogging, and walking as the three gait types. The method is *not* based on statistical analysis of training data but rather on a general gait motion model synthesized using a computer graphics human model. This makes training (from different views) very easy and separates the training and test data completely. The method is designed to handle challenges that arise in an unconstrained scene and the method has been evaluated on different data sets containing all the important factors which such a method should be able to handle.The method performs well (both in its own right and in comparison to related methods) and it is concluded that the method can be characterized as an *invariant* method for gait description.

The method is further developed to allow video input from multiple cameras. The method can achieve real-time performance and a method for online adjustment of the background subtraction method ensures the quality of the silhouette extraction for scenes with rapid changing illumination conditions.

The quality of the foreground segmentation is important for the precision of the gait classification and duty-factor estimation. The segmentation quality could be improved in the future by extending the color based segmentation of the Codebook method with edge information directly in the segmentation process and furthermore including region based information. This would especially be an advantage in scenes with poor illumination or with video from low quality cameras.

The general motion model used to generate training data effectively represents the basic characteristics of the three gait types, i.e. the characteristics that are independent of person-specific variations. Gait may very well be the type of actions that are most easily described by a single prototypical execution but an interesting area for future work could be the extension of this approach to other actions like waving, boxing, and kicking.

The link between the duty-factor and the biomechanical properties of gait could also be an interesting area for future work. By applying the system in a more constrained setup it would possible to get camera calibrations and ground plane information that could increase the precision of the duty-factor estimation to a level were it may be used to analyze the performance of running athletes.

5.14 Acknowledgment

This work was supported by the EU project HERMES (FP6 IST-027110) and the BigBrother project (Danish Agency for Science, Technology, and Innovation, CVMT, 2007-2010).

References

[1] R.M. Alexander. Optimization and Gaits in the Locomotion of Vertebrates. *Physiological Reviews*, 69(4):1199 – 1227, October 1989.

[2] R.M. Alexander. Energetics and Optimization of Human Walking and Running: The 2000 Raymond Pearl Memorial Lecture. *American Journal of Human Biology*, 14(5):641 – 648, Sep-Oct 2002.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.

[4] S-J. Blakemore and J. Decety. From the Perception of Action to the Understanding of Intention. *Nature Reviews Neuroscience*, 2(8):561–567, 2001.

[5] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as Space-Time Shapes. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1395–1402, Washington, DC, USA, 2005. IEEE Computer Society.

[6] R.T. Collins, R. Gross, and Jianbo Shi. Silhouette-Based Human Identification from Body Shape and Gait. In *FGR '02: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 351–356, Washington, DC, USA, 2002. IEEE Computer Society.

[7] Ross Cutler and Larry S. Davis. Robust Real-Time Periodic Motion Detection, Analysis, and Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):781–796, August 2000.

[8] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior Recognition via Sparse Spatio-Temporal Features. In *2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005.

[9] P. Fihl, R. Corlin, S. Park, T.B. Moeslund, and M.M. Trivedi. Tracking of Individuals in Very Long Video Sequences. In *International Symposium on Visual Computing, Advances in Visual Computing, LNCS, Vol. 4291, Springer-Verlag Berlin/Heidelberg*, Lake Tahoe, Nevada, USA, November 6-8 2006.

[10] K. Kim, T.H. Chalidabhongse, D. Harwood, and L. Davis. Real-time Foreground-Background Segmentation using Codebook Model. *Real-time Imaging*, 11(3):167–256, June 2005.

[11] T-K. Kim and R. Cipolla. Canonical Correlation Analysis of Video Volume Tensors for Action Categorization and Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1415–1428, 2009.

[12] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning Realistic Human Actions from Movies. In *CVPR 2008: IEEE Conference on Computer Vision and Pattern Recognition*, Alaska, USA, 2008.

[13] Z. Li, Y. Fu, T. Huang, and S. Yan. Real-time Human Action Recognition by Luminance Field Trajectory Analysis. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia*, pages 671–676, New York, NY, USA, 2008. ACM.

[14] Z. Liu, L. Malave, A. Osuntugun, P. Sudhakar, and S. Sarkar. Towards Understanding the Limits of Gait Recognition. In *International Symposium on Defense and Security*, Orlando, Florida, USA, April 12-16 2004.

[15] Z. Liu and S. Sarkar. Improved Gait Recognition by Gait Dynamics Normalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):863 – 876, June 2006.

[16] Osama Masoud and Nikos Papanikolopoulos. A Method for Human Action Recognition. *Image and Vision Computing*, 21(8):729 – 743, August 2003.

[17] Eric M. Meisner, Selma Àbanovic, Volkan Isler, Linnda Caporeal R. Caporeal, and Jeff Trinkle. ShadowPlay: a Generative Model for Nonverbal Human-robot Interaction. In *HRI '09: Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction*, 2009.

[18] J. M. Montepare, S. B. Goldstein, and Annmarie Clausen. The Identification of Emotions from Gait Information. *Journal of Nonverbal Behavior*, 11(1):33–42, 1987.

[19] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Courier Dover Publications, Mineola, NY, USA, 1998.

[20] A. Patron and I. Reid. A Probabilistic Framework for Recognizing Similar Actions using Spatio-Temporal Features. In *18th British Machine Vision Conference*, 2007.

[21] A. Patron, E. Sommerlade, and I. Reid. Action recognition using shared motion parts. In *Proceedings of the Eighth International Workshop on Visual Surveillance 2008*, October 2008.

[22] Yang Ran, Isaac Weiss, Qinfen Zheng, and Larry S. Davis. Pedestrian Detection via Periodic Motion Analysis. *International Journal of Computer Vision*, 71(2):143 – 160, Februar 2007.

[23] Neil Robertson and Ian Reid. Behaviour Understanding in Video: A Combined Method. In *10th IEEE International Conference on Computer Vision*, pages 808–814, Oct 2005.

[24] C. Schüldt, I. Laptev, and B. Caputo. Recognizing Human Actions: a Local SVM Approach. In *ICPR '04: Proceedings of the 17th International Conference on Pattern Recognition*, pages 32–36. IEEE Computer Society, 2004.

[25] M. Svenstrup, S. Tranberg, H.J. Andersen, and T. Bak. Pose Estimation and Adaptive Robot Behaviour for Human-Robot Interaction. In *International Conference on Robotics and Automation*, Kobe, Japan, 2009.

[26] J.B. Tenenbaum, V. de Silva, and J.C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319 – 2323, December 2000.

[27] A. Veeraraghavan, A.K. Roy-Chowdhury, and R. Chellappa. Matching Shape Sequences in Video with Applications in Human Movement Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1896 – 1909, December 2005.

[28] Paul Viola, Michael J. Jones, and Daniel Snow. Detecting Pedestrians Using Patterns of Motion and Appearance. *International Journal of Computer Vision*, 63(2):153 – 161, July 2005.

[29] Stefan Waldherr, Roseli Romero, and Sebastian Thrun. A Gesture Based Interface for Human-Robot Interaction. *Autonomous Robots*, 9(2):151–173, 2000.

[30] L. Wang, T. N. Tan, H. Z. Ning, and W. M. Hu. Fusion of Static and Dynamic Body Biometrics for Gait Recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(2):149–158, 2004.

[31] Michael W. Whittle. *Gait Analysis, an Introduction*. Butterworth-Heinemann Ltd, 2001.

[32] C.Y. Yam, M.S. Nixon, and J.N. Carter. On the Relationship of Human Walking and Running: Automatic Person Identification by Gait. In *International Conference on Pattern Recognition*, 2002.

[33] Hee-Deok Yang, A-Yeon Park, and Seong-Whan Lee. Human-Robot Interaction by Whole Body Gesture Spotting and Recognition. In *International Conference on Pattern Recognition*, 2006.

Chapter 6

2D Human Pose Estimation

This chapter consists of the paper "Pose Estimation of Interacting People using Pictorial Structures" [A]. The paper presents a method that address the problems of complex occlusions between interacting people and occlusions from other foreground objects. The method also introduces a dynamic weighting between the edge and appearance information used for detections of body parts in the pictorial structures framework.

References

A. P. Fihl and T.B. Moeslund. Pose Estimation of Interacting People using Pictorial Structures. In *IEEE International Conference on Advanced Video and Signal-Based Surveillance, Boston, MA, USA*, August 2010

Pose Estimation of Interacting People using Pictorial Structures

Preben Fihl and Thomas B. Moeslund

Abstract

Pose estimation of people have had great progress in recent years but so far research has dealt with single persons. In this paper we address some of the challenges that arise when doing pose estimation of interacting people. We build on the pictorial structures framework and make important contributions by combining color-based appearance and edge information using a measure of the local quality of the appearance feature. In this way we not only combine the two types of features but dynamically find the optimal weighting of them. We further enable the method to handle occlusions by searching a foreground mask for possible occluded body parts and then applying extra strong kinematic constraints to find the true occluded body parts. The effect of applying our two contributions are show through both qualitative and quantitative tests and show a clear improvement on the ability to correctly localize body parts.

6.1 Introduction

Recent research within computer vision has provided significant improvements for estimating the pose of humans in both still images and in video [9]. A good description of the configuration of the human body enables many interesting applications within *e.g.* video surveillance, motion capture, video indexing, and human-computer interaction. Estimating the pose of a single person in complex scenarios is however still a challenging problem and pose estimation of two interacting people is therefore rarely investigated. Estimating the pose of interacting people will significantly increase the level of applicability and for example give way for even more powerful action recognition within automatic video surveillance.

In this paper we present solutions to some of the challenges that arise when doing pose estimation of interacting people in video sequences (see figure 6.1). Our approach is based on pictorial structures and builds on the work of Felzenszwalb and Huttenlocher [3] and Ramanan *et al.* [12]. Approaches based on pictorial structures have presented very good results on human pose estimation due to their effective combination of image data and the kinematic constraints that apply to the human body. Body parts are detected using some kind of appearance model. [3] use simple binary image features extracted by background subtraction. [12] introduces person-specific appearance models based on colors that are initialized by detecting a characteristic pose. In [4, 11] the color-based appearance models are combined with edge information in an iterative approach starting from rather weak edge detections and then refining the detections with color based appearance. [1] take a slightly different approach and use log-polar histograms of gradient orientations to detect body parts.

In situations with interacting people, the person-specific appearance models of [12] provide a good basis for representing the individual body parts. Later work, like [4, 11], improve the appearance models by also including edge information. Compared to [4, 11] however, we combine appearance and edge information based on the local quality of the appearance feature, thereby not only combining the two types of features but dynamically finding the optimal weighting of them.

The pictorial structures in their original form are known to have problems with occlusions. The kinematic constraints between body parts are represented by a tree-shaped graph with edges of the tree representing the joints connecting body parts. To allow for efficient inference in this tree it was constructed as an acyclic graph which limited the model from capturing the additional dependencies between body parts that inevitably lead to self-occlusions. Different approaches have been taken to directly model such dependencies. [13] model self-occlusions directly by adding occlusion-sensitive likelihoods to the tree-model. [14] use multiple tree-models to represent the additional dependencies between body parts and then learn the internal weighting of these models. [7] propose a non-tree based approach to deal with self-occlusions without modeling the dependencies between body parts directly. The method maximizes the area of foreground covered by the body parts and incorporates

Figure 6.1: Our approach estimates the human pose during interactions and occlusions without assumptions regarding possible body configurations.

kinematic constraints between body parts, edge-based image features, and color symmetry between left and right limbs.

The possible complex occlusions of interacting people can, however, not easily be modeled directly and the occlusion modeling becomes even harder when, in addition, occlusions from other foreground objects can occur, *e.g.* a person carrying a briefcase. Shape-based methods like [10, 6] and the maximum foreground covering of [7] will also face problems when trying to explain a foreground region consisting of multiple people and possibly other objects. Building shape databases covering unrestricted interactions is infeasible and maximizing the coverage of foreground will also cause errors when objects are present.

We address the problems of occlusions by first doing foreground segmentation and then using the foreground mask to search for occluded body parts independently of image evidence. This will naturally result in many noisy body part detections but by imposing extra strong kinematic constraints on the occluded body parts we end up with good pose estimates.

The rest of this paper will first briefly review the pictorial structures framework and then describe the method used for foreground segmentation. Sections 6.4 and 6.5 will describe the two main contributions of the paper, *i.e.*, the dynamic weighting of appearance features versus edge features and the occlusion handling using foreground segmentation. Section 6.6 presents results and section 6.7 concludes the paper.

6.2 Pictorial structures framework

We use the same formulation of the pictorial structure framework as in [12]. Body parts are detected using a discriminative color classification and rectangular image patches that represent (x, y)-coordinates and the orientation θ. These body part detections are combined by a set of pairwise constraints. This can be represented in

a tree-shaped graph which makes MAP estimation computationally efficient. The formal description of the pictorial structures is as follows. The posterior P of the body configuration L given an image I and an appearance model Λ is found using equation 6.1.

$$P(L|I,\Lambda) \propto \exp\left(\sum_{(i,j)\in G} \Psi(l_i,l_j) + \sum_i \Phi(l_i|I,\Lambda)\right) \tag{6.1}$$

$\Psi(l_i,l_j)$ is the pairwise potential that corresponds to the kinematic constraints between body parts (l) represented in the tree-shaped graph G. We define the pairwise potentials to be uniform within a bounded range. This gives increased flexibility in the relative positions of the body parts resulting in a good body model without modeling the scale of each body part. $\Phi(l_i|I,\Lambda)$ is the unary potential that describes the dissimilarity between the image evidence at a given location and the image patch of body part l_i. In [12] Φ is based on edge templates for initialization and on color-based appearance for subsequent detection. [11, 4] extend Φ to combine edges and color-based appearance and we extend this even further by including a dynamically adjusted weighting between edges and color-based appearance. This will be described further in section 6.4.

The appearance models are person-specific and automatically initialized on isolated persons. This is done by restricting the initialization to a characteristic walking pose which is easy to detect and gives a good view of the individual body parts. This is sometimes considered to be a restriction, for example when working with video from tv-shows ([1, 4]), but when working with real-life video acquired from, *e.g.*, surveillance cameras then people that appear in the scene are most often walking into the scene. The restriction of the walking pose is minimal and by using this initialization we get a good description of the appearance of individuals while they are apart which significantly improves the pose estimation during interaction. Initialization will be attempted on new foreground regions with areas corresponding approximately to human size. A successful initialization will result in a new appearance model and the number of people in the scene can be managed accordingly.

Inference in the tree is done by message-passing (the sum-product algorithm) and by sampling many possible body configurations we get the posterior as probability distributions over position and orientation over each body part.

6.3 Foreground segmentation

The detection of body parts in images inevitably leads to many false positives. Applying the kinematic constraints between body parts effectively handles parts of this problem but we still want to allow rather arbitrary poses that occur in real-life video and which are often important poses when recognizing the actions of people; this could for example be a person falling to the ground or jumping back to avoid

a passing car. To increase the precision of the pose estimation we therefore first apply a foreground segmentation to reduce the search space for body parts. In [4] the search space is reduced in two steps by first applying a HOG-based upper-body detector and next doing a soft segmentation of body parts within the upper-body detections. However, doing detection of individual persons is prone to errors when people are interacting closely, *e.g.*, when hugging. We therefore take a different approach and reduce the search space by background subtraction. In this way we reduce the search space to foreground regions and we find the overall scale of a person at the same time. Including all foreground regions in the search space also allows us to search for body parts that are occluded by other foreground objects which will be detailed in section 6.5.

Background subtraction requires a static camera and is therefore not applicable in all scenarios, but the majority of cameras that are observing people are static cameras (typically surveillance cameras). This means that the use of efficient background subtraction is not a limiting factor for a great number of real-life application.

We use the Codebook background subtraction method presented in [8] and expanded in [5]. This method has been shown to be robust in handling both foreground camouflage and shadows. This is achieved by separating intensity and chromaticity in the background model. Moreover, the background model is multi-modal and multi-layered which allows it to model moving backgrounds such as tree branches and objects that become part of the background after staying stationary for a period of time. To maintain good background subtraction quality over time it is essential to update the background model and [5] describes two different update mechanisms to handle rapid and gradual changes respectively. By using this robust background subtraction method we achieve good segmentation results in real-life outdoor scenes.

6.4 Dynamic optimization of appearance model

As described earlier our unary potential Φ of the pictorial structure framework is a combination of edge information and color-based appearance but rather than using a static combination of the two we dynamically evaluate the local quality of the color-based appearance and thereby find the optimal weighting of the two types of information. The unary potential in our approach becomes

$$\Phi(l_i|I,\Lambda) = \alpha(x,y,\theta)\cdot A(l_i) + E(l_i) \tag{6.2}$$

where α is a local weighting factor, A is the color-based appearance, E is the edge-based information and the body parts l_i are described by (x,y)-position and orientation θ.

Like in [12] the color-based appearance works by first learning a color classifier for each body part from the initial walking pose. Subsequent frames are then classified to find pixels possibly belonging to each body part. By convolving this body part mask with a general rectangular body part model we find $A(l_i)$ for all (x,y,θ). By

a similar approach we convolve general edge-based body part templates with edge detections of I to find $E(l_i)$ for all (x, y, θ).

To combine A and E we calculate the weighting factor α that essentially describes the local quality of the color-based appearance. In a large, uniformly colored region of the image, like a frontal torso, $A(l_i)$ will have a uniform response for a similar-colored smaller body part, like a lower arm (see figure 6.2). [4, 11, 12] try to strengthen the descriptive power of the color-based appearance by making it discriminative, *i.e.*, measuring $A(l_i)$ by both positively classified pixels within the rectangular body part model and negatively classified pixels surrounding the model. This approach is effective for a body part with good color-contrast against it's background, but since there are no negatively classified pixels surrounding a lower arm in front of a similarly-colored torso, then the approach tends to over-estimate the likelihood of arms being down the side of the torso (or at the edges of any other similarly-colored region). To overcome this problem we calculate α by measuring the similarity between the color of the rectangular body part model and the color of the area immediately around it, *i.e.*,

$$\alpha = w \cdot \frac{C_s(l_i)}{S(l_i)} \tag{6.3}$$

where $S(l_i)$ is the surrounding area with a width corresponding to half the width of the body part and $C_s(l_i)$ is the surrounding area of l_i that is positively classified by the color classification. w is a scaling factor that defines the overall importance of the color-based appearance versus the edge-based templates.

This relatively simple weighting ensures that the strongest image evidence at any given image location (either color or edges) is given the highest weight which is especially important when people with multi-colored or similar-looking clothes are interacting. Figures 6.2 and 6.3 illustrate how the importance of color-based appearance and edge information are dependant on the characteristics of the image. Edge information is most reliable in the context of pose estimation in figure 6.2 whereas the appearance information is more reliable in figure 6.3. A static combination of the two features will not perform as well as our weighting based on the local quality of the color-based appearance.

6.5 Occlusion handling

When dealing with interacting people and real-life video it is crucial to handle occlusions efficiently. It is not sufficient to recover from poor pose estimates after occlusions (which is ensured by doing body part detections in every frame); it is also necessary to do good pose estimation during occlusions since for example many interactions carry important information.

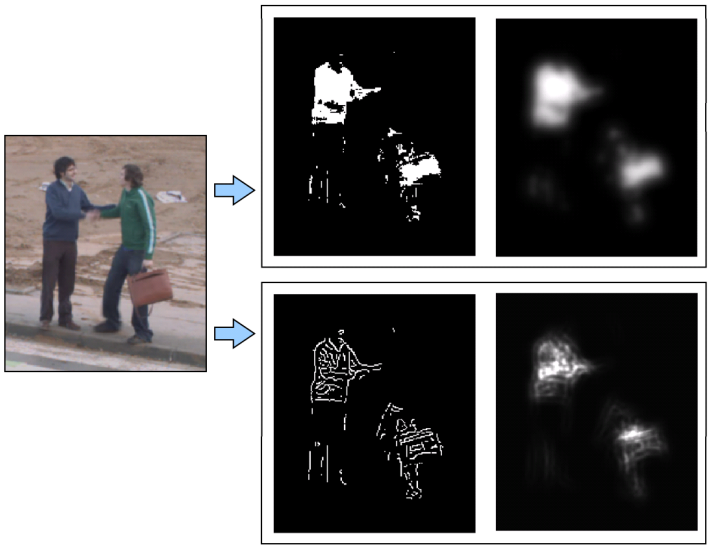


Figure 6.2: The importance of edge information in pose estimation, here exemplified by the pose estimation of the lower arm of the left person. The input image to the left is processed to show both the color-based appearance response (*middle-top*) and the edge-based response (*middle-bottom*). The right images show the probability distribution derived from either color-based appearance (*right-top*) or the edge detections (*right-bottom*). Here, the edge information gives significantly more accurate pose estimates compared to the appearance-based information.

As previously stated, the general problem of occlusions (*i.e.*, both self-occlusions, inter-person occlusions, and occlusions by other foreground objects) cannot be modeled efficiently by extending or adding constraints on the kinematic model of the pictorial structures. Instead we make use of the foreground segmentation to search for body parts that are occluded (and thereby do not directly have support from image evidence).

Initially we assume that everything within the foreground mask F that does not correspond to the color-based appearance of a body part is potentially occluding that body part (see figure 6.4). In terms of binary images this is formulated as $O = F \oplus C(l_i)$, where $C(l_i)$ are the pixels classified as belonging to body part l_i. We convolve the binary mask O with the rectangular body part model of l_i to find possible occluded body parts. This step is equivalent to the convolution with general rectangular body part models involved in the calculation of the appearance-based potential $A(l_i)$ (see section 6.4). However, since the occluded body parts are not directly supported by image evidence but rather based on the assumption that all pixels in O potentially occlude body part l_i, we cannot directly convert the response

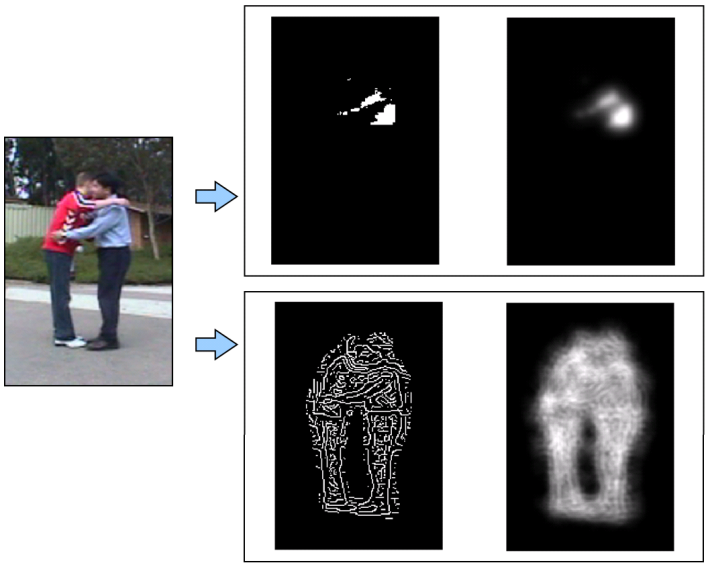


Figure 6.3: The importance of color-based appearance in pose estimation, here exemplified by the pose estimation of the lower arm of the right person. The input image to the left is processed to show both the color-based appearance response (*middle-top*) and the edge-based response (*middle-bottom*). The right images show the probability distribution derived from either color-based appearance (*right-top*) or the edge detections (*right-bottom*).The appearance-based information is more accurate than the cluttered edge detections.

of the convolution to a meaningful unary potential (which is done for $A(l_i)$). To include the occluded body parts in the pictorial structure framework we therefore assign them a unary potential Φ based on the lower limb (l_l) that they connect to, *i.e.*,

$$\Phi(l_o) = \beta \cdot \Phi(l_l) \tag{6.4}$$

where β is a scaling factor

This second set of body part detections are naturally very noisy since the binary mask O contains much more than actual occluding pixels (see figure 6.4). However, by discarding the detections that do not comply with the kinematic constraints of the original body part detections we find the possibly occluded body parts, see figure 6.5. Without image evidence to support the occluded body part detections we need to enforce stronger kinematic constraints compared to body parts detected with color-based appearance and edge information. We limit the search for occluded body parts to upper legs and upper arms which allows us to reduce the occluded body part to those that are connected to both a torso and respectively a lower leg or a lower arm.

Figure 6.4: *Top-left:* the input image. *Top-right:* the foreground mask F. *Bottom-left:* the pixels classified by the upper-leg appearance model C. *Bottom-right:* the mask that is searched for possible occluded upper legs O, *i.e.* the foreground mask except the mask from the appearance classification.

Lower limbs cannot be constrained as easily, and we therefore merely allow upper arms and upper legs to connect to a null-segment with low unary potential to simulate occluded lower arms and lower legs. The resulting set of occluded body parts are included in the inference process with their unary potential just as normal body parts.

Another issue to deal with in the context of occlusion handling is double-counting of image evidence, *i.e.*, situations where the pose estimation results of, for example, two legs explain the same image data. When directly maximizing $P(L)$ then both legs or both arms of the body model will tend to explain the same high-probability image region. We apply an approach similar to [12] and iteratively find the limb (*i.e.*, torso+head, upper+lower arm, or upper+lower leg) with the highest probability and for each iteration remove the samples that overlap completely with the selected limb. Like in [12] we use the mean shift algorithm [2] to find the best mode of the posterior distribution. At this stage we furthermore augment the probability with a measure of the degree of overlap with other limbs of the same type. This is done to increase the amount of image data explained by the final pose estimate.

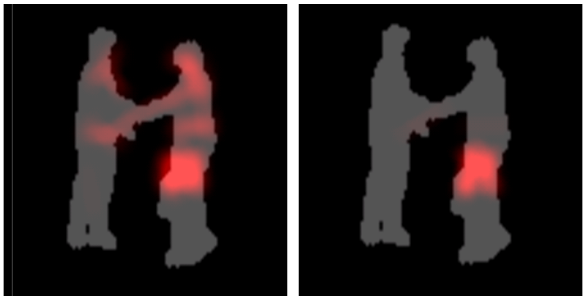


Figure 6.5: *Left:* the possible occluded upper legs found by searching O. *Right:* the remaining occluded upper legs after applying the strong kinematic constraints.

6.6 Results

To demonstrate the effect of our dynamic weighting of the color-based appearance and our approach to occlusion handling we show both qualitative and quantitative comparisons. We compare the results of the method proposed in this paper to a parallel implementation that uses a static combination of edges and color-based appearance and no explicit occlusion handling. Both implementations use the same foreground segmentation to reduce the search space for body parts. Edges and color-based appearance are weighted equally in the static combination of the two. The scaling factors w and β for the proposed method are found empirically and set to 2 and 1/4 respectively. [1]

Figures 6.6 and 6.7 show comparative results on two natural outdoor video sequences. The comparison clearly shows that the dynamic weighting of the color-based appearance increases the ability to correctly localize body parts that have similar appearance to their background (*e.g.*, the lower arms in front of the torso in figure 6.6 column b and c and figure 6.7 column d). The comparison also shows the benefit of our occlusion handling. Our approach correctly localizes occluded and partially occluded body parts that would otherwise cause whole limbs to be missed (*e.g.*, the upper leg occluded by a briefcase in figure 6.6 column a, b, and d). The occlusion handling also increases the accuracy of pose estimates of partially occluded body parts (*e.g.*, the lower legs in figure 6.7 column b and the torso in figure 6.7 column c).

To quantitatively demonstrate the effect of our approach we have manually annotated the ground truth poses in 75 frames from the video sequence of figure 6.6 and in 50 frames from the video sequence of figure 6.7 (all frames sampled randomly).

[1] The implementation is based on the code published by Ramanan (www.ics.uci.edu/~ dramanan/papers/pose/).

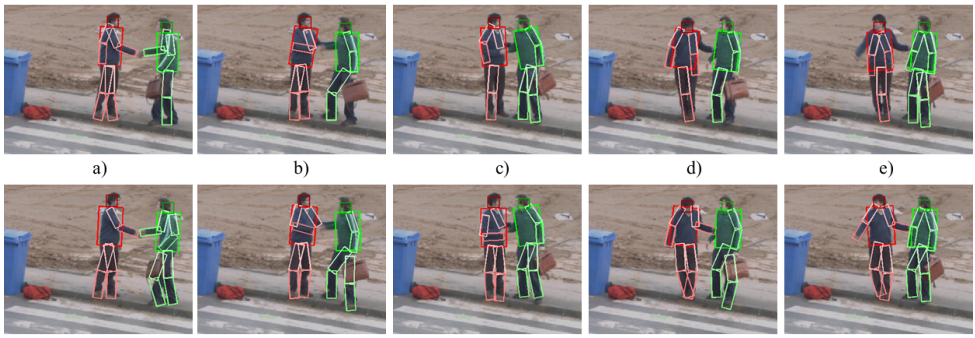


Figure 6.6: Example results. *Top row:* results using static weighting between edges and the color-based appearance and no occlusion handling. *Bottom row:* results using the method presented in this paper with dynamic weighting of edges and color-based appearance and with occlusion handling.

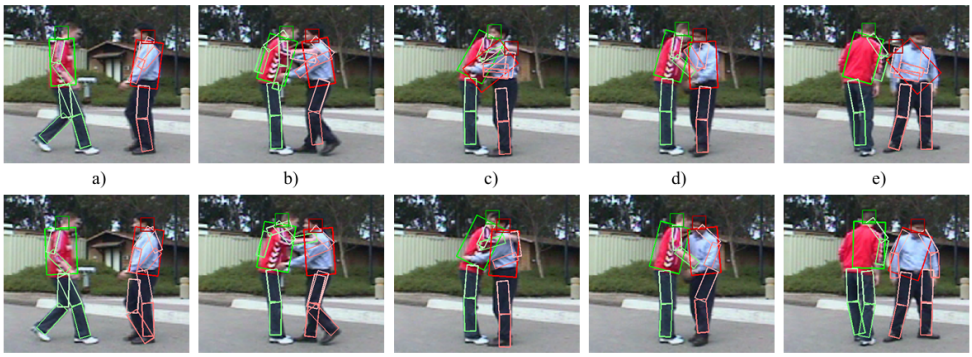


Figure 6.7: Example results. *Top row:* results using static weighting between edges and the color-based appearance and no occlusion handling. *Bottom row:* results using the method presented in this paper with dynamic weighting of edges and color-based appearance and with occlusion handling.

In addition to the ground truth pose each body part is also marked as being either occluded or visible. If more than half of a body part is not visible, then it is marked as occluded. This allows us to specify the results with respect to occluded body parts.[2]

We use the criterion for correctly located body parts defined in [4] and also used in [1]. A body part is correctly localized if the endpoints of its segment lie within 50% of the ground truth length from their true positions. Table 6.1 lists the pose estimation results.

[2]The video data and ground truth annotations is available from the authors website

All body parts	**Total**	Torsos	Arms	Legs	Heads
Dynamic weighting and occlusion handling	**72.9**	99.6	49.9	89.5	58.7
Static weighting and no occlusion handling	**62.9**	95.2	43.7	74.8	48.8

Occluded body parts	**Total**	Torsos	Arms	Legs	Heads
Dynamic weighting and occlusion handling	**54.2**	-	45.7	64.5	25.0
Static weighting and no occlusion handling	**40.5**	-	33.6	46.7	75.0

Table 6.1: Percentages of correctly localized body parts. The results compare two setups and specify the results on the body parts that were marked as occluded. Setup 1) Results obtained using the dynamic weighting of edges and color-based appearance and the occlusion handling presented in this paper. Setup 2) Results obtained with a parallel implementation that uses a static weighting of edges and color-based appearance and no explicit occlusion handling. No torsos and only four heads were occluded in the 125 test frames. 2247 ground truth body parts were marked out of which 227 were occluded.

It is clear to see from table 6.1 that our approach improves the number of correctly localized body parts leading to better pose estimates. The percentage of correctly localized body parts increase from 62.9% to 72.9%. The increase is even larger when looking solely on the ability to localize occluded and partly occluded body parts. The percentage of correctly localized occluded body parts increase from 40.5% to 54.2%.

Figure 6.8 shows examples of body parts that are not localized correctly due to foreshortening and ambiguity in both edge information and color-based appearance. Foreshortening is to some extend implicitly handled by the mean shift procedure when finding the best modes of the pose probability distributions. The resulting pose is a local average over joint positions and according to [12] this smoothing captures foreshortening with results comparable to a direct search over body parts lengths. However, the smoothing does not cope with severe foreshortening and such body parts tend to give pose estimation errors (see figure 6.8 left). Ambiguity in both edge information and color-based appearance most often occur when an arm is in front of the torso, *e.g.* figure 6.8 (right). Edges from the arms together with edges from the torso may result in relatively high probabilities for arm poses along the sides of the torso. The color-based appearance may have poor quality, and hence little effect, since arms and torsos often have similar appearance. This issue can be effectively dealt with by including temporal information.

(www.cvmt.dk/∼pfa/pose/).

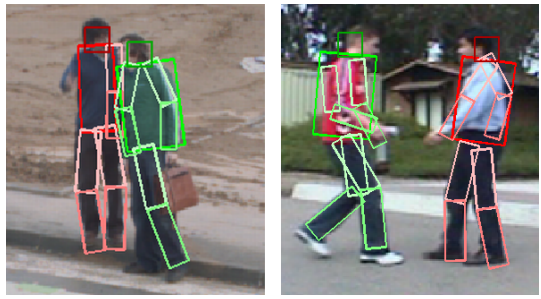


Figure 6.8: Examples of missed or erroneously localized body parts. *Left:* the arm of the left person is not detected due to severe foreshortening. *Right:* the combination of edges around the arm and the similar appearances of arm and torso cause the wrong localizations of arms along the sides of the torso (true for both persons). The same type of error is seen in figure 6.7 column a.

6.7 Discussion

In this paper we present an approach based on the pictorial structures framework that efficiently handle some of the challenges of doing pose estimation on interacting people. The body part detections are based on a combination of color-based appearance and edge information. We dynamically measure the quality of the color-based appearance and weight color and edge information in accordance with this quality measure to get the optimal combination at all times. To enable the method to handle the increased number of occlusions that typically occur during interaction we search the foreground mask for possible occluded body parts and then apply extra strong kinematic constraints to find the true occluded body parts. The effect of applying our two contributions are shown through both qualitative and quantitative tests and show a clear improvement on the ability to correctly localize body parts.

The level of accuracy needed in pose estimation to allow for recognition of actions and interactions is not clearly defined. However, with 72.9% correctly localized body parts in a natural outdoor scene with many occlusions our approach delivers a good basis for such recognition.

The appearance models are initialized in a characteristic walking pose which is a reasonable approach in the context of action recognition and real-life video. However, all subsequent pose estimation results are dependant on this initialization which can cause problems and a more general initialization would be preferred.

The current pose estimation is done frame-by-frame and it would surely be advantageous to include the temporal information in a tracking scheme to improve the pose estimation results further.

References

[1] M. Andriluka, S. Roth, and B. Schiele. Pictorial Structures Revisited: People Detection and Articulated Pose Estimation. In *CVPR*, 2009.

[2] Dorin Comaniciu and Peter Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis. *PAMI*, 24(5), 2002.

[3] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial Structures for Object Recognition. *Int. Jour. Computer Vision*, 61(1), Jan 2005.

[4] Vittorio Ferrari, Manuel J. Marn-Jimnez, and Andrew Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, 2008.

[5] P. Fihl, R. Corlin, S. Park, T.B. Moeslund, and M.M. Trivedi. Tracking of Individuals in Very Long Video Sequences. In *International Symposium on Visual Computing, Advances in Visual Computing, LNCS, Vol. 4291, Springer-Verlag Berlin/Heidelberg*, Lake Tahoe, Nevada, USA, November 6-8 2006.

[6] Dariu M. Gavrila. A Bayesian, Exemplar-Based Approach to Hierarchical Shape Matching. *PAMI*, 29(8), 2007.

[7] H. Jiang. Human Pose Estimation Using Consistent Max-Covering. In *ICCV*, 2009.

[8] K. Kim, T.H. Chalidabhongse, D. Harwood, and L. Davis. Real-time Foreground-Background Segmentation using Codebook Model. *Real-time Imaging*, 11(3), June 2005.

[9] T.B. Moeslund, A. Hilton, and V. Krüger. A Survey of Advances in Vision-Based Human Motion Capture and Analysis. *Jour. of Computer Vision and Image Understanding*, 104(2-3), 2006.

[10] G. Mori and J. Malik. Recovering 3d Human Body Configurations Using Shape Contexts. *PAMI*, July 2006.

[11] D. Ramanan. Learning to Parse Images of Articulated Objects. In *Neural Info. Proc. Systems*, Dec 2006.

[12] D. Ramanan, D. A. Forsyth, and A. Zisserman. Tracking People by Learning their Appearance. *PAMI*, 29(1), Jan 2007.

[13] Leonid Sigal and Michael J. Black. Measure Locally, Reason Globally: Occlusion-sensitive Articulated Pose Estimation. In *CVPR*, 2006.

[14] Yang Wang and Greg Mori. Multiple Tree Models for Occlusion and Spatial Constraints in Human Pose Estimation. In *ECCV*, 2008.

Chapter 7

Multi-view Human Pose Estimation

This chapter consists of the technical report "Full Body Pose Estimation During Occlusion using Multiple Cameras" [A]. The report presents work towards an integration of low level information from multiple cameras to effectively deal with heavy occlusions in pose estimation. The unconstrained camera setups of typical surveillance scenarios and the complex inter-person occlusions when people interact are the main problems addressed in this work. The report describes research in progress and is presented here to document work that takes the pose estimation of chapter 6 into a new direction.

References

A. P. Fihl and S. Cosar. Full Body Pose Estimation During Occlusion using Multiple Cameras. In *Technical Report CVMT-10-02, Laboratory of Computer Vision and Media Technology, Aalborg University, Denmark*, 2010.

Full Body Pose Estimation During Occlusion using Multiple Cameras

P. Fihl and S. Coşar

Abstract

Automatic estimation of the human pose enables many interesting applications and has therefore achieved much attention in recent years. One of the most successful approaches for estimating unconstrained poses has been the pictorial structures framework. However, occlusions between interacting people is a very challenging problem for methods based on pictorials structure as for any other monocular pose estimation method. In this report we present work on a multi-view approach based on pictorial structures that integrate low level information from multiple calibrated cameras to improve the 2D pose estimates in each view. The proposed method is shown to work under heavy occlusions but does not improve the pose estimates in the non-occluded cases in its current form.

7.1 Introduction

Automatic estimation of the human pose enables many interesting applications and has therefore achieved much attention in recent years. Accurate pose estimations can give a good description of the actions being performed in a video and hence be used for *e.g.*, automatic video surveillance, human-computer interaction or automatic video annotation. In this report we present a method to do full body pose estimation by combining information from multiple cameras to deal with the problem of occlusions between people.

One of the most successful approaches for estimating unconstrained poses has been the pictorial structures framework [7] which has been improved and extended in a number of works [2, 8, 18]. One of the main challenges for human pose estimation is the inherent problem of occlusions, especially self occlusions, i.e. one body part occluding another body part of the same person. Several methods have been proposed to deal with this problem by modeling the self occlusions directly in the body model [19, 20] or by utilizing a foreground mask and maximizing the area of foreground covered by body parts [15].

The problem of occlusions increases when multiple people interact. [6] proposes a multi-person pictorial structures model and estimate the front-to-back ordering of people to find the probability of occlusions between people. [9] also addresses the problem of occlusions by other foreground objects and propose to detect occluded body parts by pruning the foreground mask into a mask of possible occlusions and then include these detections in the inference process of the pictorial structures.

All of these approaches use a single viewpoint to provide a 2D pose estimate but severe occlusions will always cause problems for monocular methods in general. Combining information from multiple cameras can help to solve this issue and a number of multi-view methods for reconstruction of the human pose have been proposed in recent years, based on for example visual hull [4, 12] or skeleton models [3, 5].

The multi-view methods range from carefully calibrated studio setups to surveillance setups with the placement of the cameras determined by the environment. The studio setups usually deal with human pose estimation in 3D and the cameras are arranged for that purpose specifically. [3, 4, 5, 12, 17] are examples of such methods and [11] presents a representative multi-view data set for human pose estimation and action recognition with references to related methods and data sets. The surveillance-like setups utilize multiple overlapping views in unconstrained environments and these methods typically deal with the problem of tracking people through occlusions rather than pose estimation, like in [10, 16, 21]. The video data in surveillance setups does rarely cover the scenes as thoroughly as studio setups and the video is not specifically captured with pose estimation as a goal making it a difficult task. However, in this report we present a method to combine information from multiple overlapping cameras in a surveillance-like setup into 2D full body pose estimates thereby effectively dealing with the problem of occlusions in single-view approaches.

Other approaches address a similar problem. [13] do full body pose estimation by combining the body part likelihoods from two cameras in an iterative approach. [14] use silhouette based shape matching in each view and then project the best matching shape models to the other views where a multi-view matching score is calculated. Our method also integrates low level information, like the body part likelihoods, but we build on the pictorial structures framework which has shown good results in single-view approaches and we do the integration in a non-iterative way. The use of the pictorials structures framework also means that we are not limited to a set of poses represented in a database as in [14].

Our multi-view pose estimation process will not attempt to generate a 3D pose estimate. Instead, our method will integrate information from multiple views to get improved 2D pose estimations. There are no restrictions on the camera setup which fits well with the typical surveillance scenario. The method will integrate information from a variable number of cameras and does not require a person or a body part to be visible in all cameras.

The rest of the report is organized as follows. First, section 7.2 gives a brief description of the single-view pictorial structures framework. Section 7.3 then describes the integration of pose estimation from multiple cameras. Section 7.4 presents results using the proposed method which are discussed in section 7.5. Finally, section 7.6 concludes the report.

7.2 Single-view pictorial structures framework

Our approach extends the single-view pose estimation method presented in [9] which build on the pictorial structures framework of [18]. The single-view pose estimation method in [9] will be summarized next.

The pose estimation is based on detections of individual body parts which are combined into body configurations under kinematic constraints in a pictorial structures framework. The body model contains ten body parts but only six types of body parts are detected since left and right limbs are assumed to have the same appearance and shape. The body part types are heads, torsos, upper arms, lower arms, upper legs, and lower legs (see figure 7.1).

A robust background subtraction is first applied to limit the search space for body parts to foreground regions. The body parts are detected using both person-specific appearance models and generic edge-based shape models. The two types of information are combined with a dynamic weighting dependant on the local quality of the appearance information.

The initialization of the appearance models is done by detecting isolated people in a characteristic walking pose. The detection of body parts in the initialization process is done only with edge-based detectors but constraining the initialization phase to one characteristic pose ensures good detections despite the relatively weak

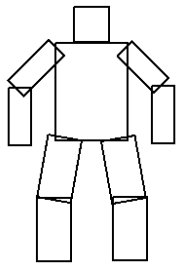


Figure 7.1: Pictorial structure representation that models the human body.

detectors. Initialization can alternatively be done by clustering edge-based body part detections [18]. This requires a set of frames for initialization but does not require people to be in the characteristic walking pose.

A set of pairwise kinematic constraints are applied to the individual body parts detections. These are represented as a tree-structured model which allow for efficient calculation of the posterior probabilities. At this stage the number of visible body parts is unknown and a set of body configurations with one arm and one leg is drawn from the posterior of the pictorial structure. The true modes of the posterior (representing the poses of visible body parts) are then found using a mean shift approach to produce the final pose estimate.

7.3 Multi-view pose integration

The body part detection of the single-view pose estimation method produces a probability map for each type of body part. By combining these probability maps from multiple cameras we will get a fusion of low level data that will ideally improve the results of the inference in the tree-structured body model and the subsequent estimation of the final pose, especially when body part detections from one view are poor as a result of occlusion. Figure 7.2 shows how our approach fits into the single-view approach.

To enable the integration of the body part probability maps we use calibrated cameras. This allows us to calculate the projections of probability maps from different cameras into a common world coordinate system (illustrated in figure 7.3). By sampling new body parts from this 3D space and projecting them back to each view we get new enhanced 2D body part probability maps. In essence, our multi-view pose estimation method combines body part probability maps from multiple cameras and generates improved probability maps for each camera. The rest of the pose estimation proceeds as in the single-view method.

The body part probability maps are confined by the foreground masks which ensures that the overlap between their projections in world coordinates is within a small

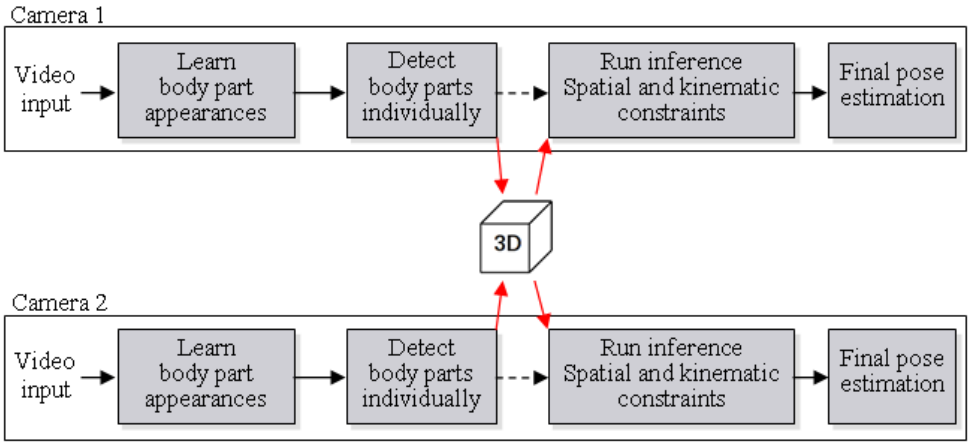


Figure 7.2: A block diagram that shows how our approach fits into the single-view approach for combining information from two cameras.

volume. However, when multiple people interact the foreground always contain more than one person, so the foreground mask merely defines a region of interest and not the silhouette of each person.

The probability maps express the probability of a body part given its 2D orientation. When projecting these into 3D we would ideally want to know the third rotation angle (out of the image plane) to ensure that the probabilities are only combined with the corresponding 3D orientation from the other views. This is however not possible. One may try to combine all possible 3D orientations for a given 2D orientation with all 2D orientations in the other views. Rather than such a complex approach, we change representation to joints instead of body parts, *i.e.*, we transform each body part probability map into two joint probability maps. For instance, upper arm probability map is transformed into shoulder and elbow probability maps. By describing a body part by its two end points we get the locations of the corresponding joints. Joint probability maps can now be generated by letting the probability of both joints be the same as the probability of the corresponding body part.

The joint probabilities $P(J)$ are independent of orientation (θ) so we take the maximum of body part probabilities $P(Bp)$ over all orientations:

$$P(J|x,y) = max(P(Bp|x',y',\theta)), \forall\theta \tag{7.1}$$

where x and y are image coordinates of the joint and x' and y' are the image coordinates of the corresponding body part center.

With known camera calibrations we can now project the non-zero joint probabilities into the 3D world space where it will represent a volume. This volume can be combined with the projections of probabilities of the same joint from other cameras.

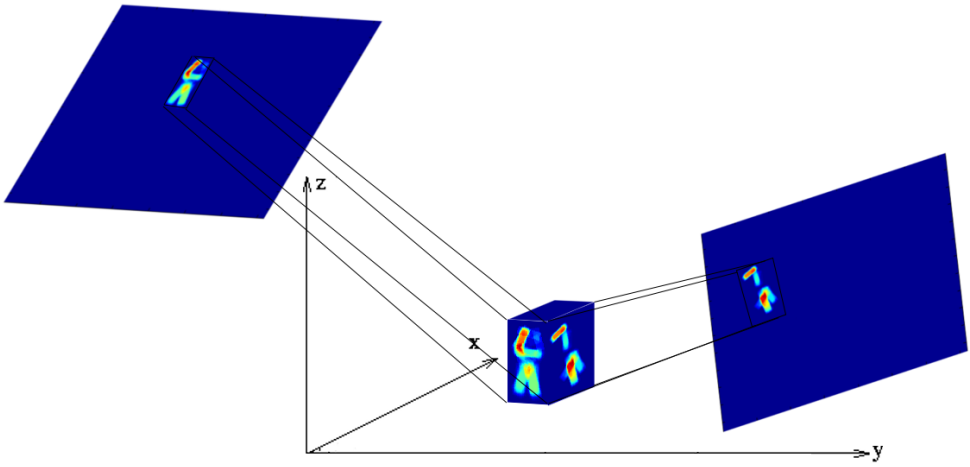


Figure 7.3: Probability maps from different viewpoints are combined by projecting them into a common world space (the illustration is not a correct perspective projection). The maps illustrate the probabilities of arms and legs with dark blue corresponding to zero probability and red corresponding to high probability.

The probabilities are added together where ever the volumes of the projections overlap.

The projection of joint probabilities corresponds to a cylindrical volume in 3D. In world space however, we want joints to be represented as spheres so we apply mean shift clustering to divide the combined volume into a set of clusters, each cluster representing a joint location, and the probability of that joint being the maximum probability of points belonging to the cluster.

This process is done for each joint individually. The goal is however to generate improved *body part* probability maps for each view, so at this stage we will connect top and bottom joints to get a body part representation, for example, we connect a shoulder joint (top joint) to an elbow joint (bottom joint) to get an upper arm sample. By back-projecting many body part samples, we can approximate the corresponding probability maps in each view.

Two joint-clusters are connected if the distance between them corresponds to the length of the appropriate body part $\pm 20\%$ of the length. Body part lengths are estimated for each person in the initialization phase. The estimation is based on a set of standard body part proportions that are scaled with the height of the person in the initialization pose. The joint-cluster are furthermore only connected if the area between the clusters project back onto the foreground mask in all views. The second condition is necessary to minimize the risk of connecting joints that do not correspond to real body parts.

One sample is drawn from each connected cluster-pair and the probability of that body part sample is the sum of the probabilities of the two joints. All these 3D body part samples are projected back onto each view where they will be connected into body configurations through inference in the tree-structured body model as in the single-view approach. The final pose estimate is also found like in the single-view case, *i.e.* using a mean shift approach to find the true modes of the posterior.

7.4 Results

This section presents the results of the preliminary tests of the proposed integration of multiple views for pose estimation.

We test the method on a video sequence from the PETS 2009 data set [1]. The data set provides video from eight calibrated cameras with people interacting in the overlapping field of view of all cameras. We generate the pose estimation results from two cameras using the single-view approach [9] and compare these results to the pose estimates found by using the proposed integration of body part probability maps from multiple views. Both sets of results use the same body part detection procedure. The multi-view approach then integrates body part probability maps but then proceeds as in the single-view approach. The comparison is based on a qualitative analysis of example frames.

The single-view approach generally performs better in the non-occluded cases (figure 7.4 rows *a* and *b*). This will be discussed in the next section. The multi-view approach performs better in the occluded cases (figure 7.4 rows *c* and *d*), especially under full occlusion (figure 7.4 column 3).

The main effect of the proposed multi-view pose integration at this stage is the ability to transfer a good pose estimate from one view into an improved pose estimate in another view where the person is heavily occluded.

7.5 Discussion

A very important difference between the single-view approach and the presented multi-view approach is the assumptions about body part proportions. The single-view approach assume fixed body part proportions in the image plane. The scale of a body part is estimated during initialization, but after that the ratio of length to width is fixed. This means that the single-view approach at this stage does not handle foreshortening at all. The flexibility of the joints in the pictorial structures model and the mean shift approach for finding the true modes of the posterior allow the single-view approach to handle some foreshortening but not very much. When estimating the poses of walking people this rarely becomes a problem but if a person were to for example point at something close to the camera the foreshortening would cause the single-view approach to fail.

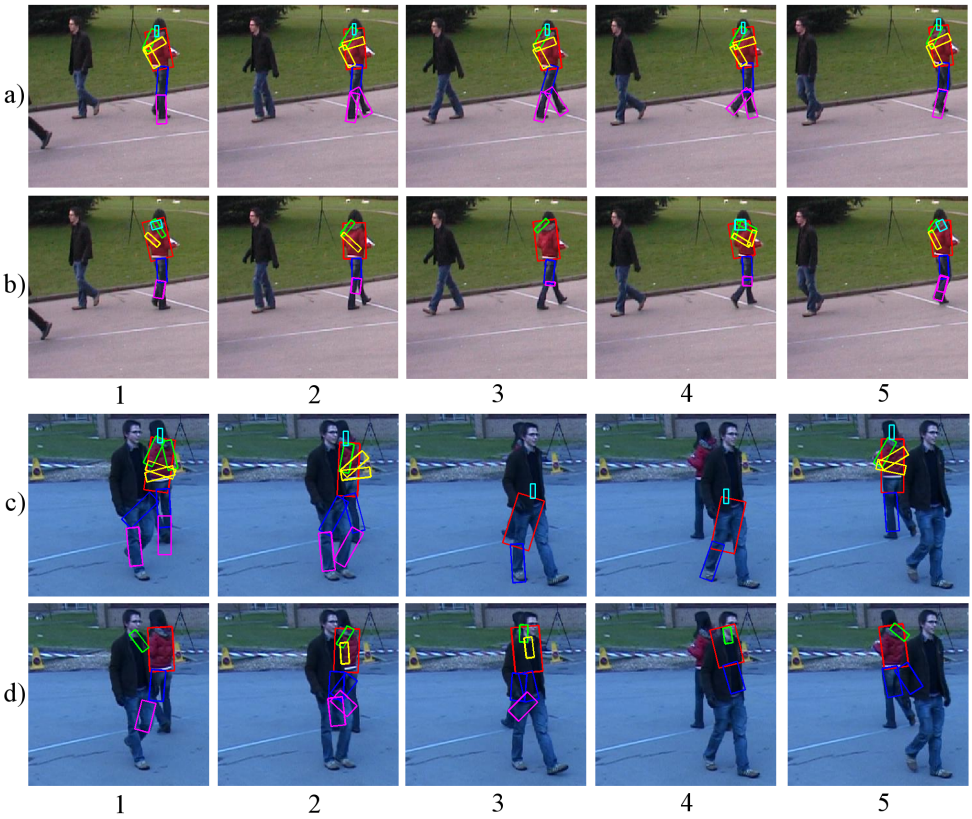


Figure 7.4: Comparison of the multi-view approach against the single-view approach. Rows a) and c) show the results of pose estimation with the single-view approach. Rows b) and d) show the results of pose estimation with the multi-view approach. Each column of the figure shows a frame from two synchronized cameras.

The multi-view approach also assumes fixed body part proportions but in the 3D world space and not in the image plane. This means that the multi-view approach can handle the extreme foreshortening of a person pointing towards the camera as long as another camera is capturing the arm without the foreshortening. Keeping the constrain on body part proportions from the single-view approach would limit the multi-view approach from an interesting capability. The results show however that lifting the constrain of body part proportions in the image plane significantly increases the noise in the body part probability maps that are generated from back projections of the 3D body parts. The connection of 3D joints into body parts does not necessarily result in a body part corresponding to a sample from one of the views (this is what allow the handling of extreme foreshortening). However, many joint-

pairs correspond to a foreshortened body part in one view also when there is no real foreshortening resulting in increased noise in the body part probability maps. This side effect significantly reduces the performance of the multi-view pose integration, an effect that is seen clearly in the non-occluded cases (figure 7.4 rows *a* and *b*).

When people undergo heavy occlusion the single-view pose estimation tend to collapse whereas the multi-view approach maintain a reasonable pose estimate. By combining the pose estimation with a person tracker it would be possible to predict when occlusions could occur and then rely on the single-view pose estimation in the non-occluded cases and include the multi-view pose integration when occlusions occur.

7.6 Conclusion

This report presents a multi-view extension to the pose estimation method of [9] based on the pictorial structures framework. Low level information about body part probabilities from multiple cameras are combined in 3D world space by utilizing calibrated cameras. The combined probabilities are projected back into each view to generate improved 2D pose estimates. Preliminary results indicates that the proposed method improves pose estimates under heavy occlusion compared to the single-view approach but it performs worse when there is no occlusion.

A number of alternatives to individual steps in the method could be explored further. The most interesting one would be to build the tree-structured body model in 3D rather than in 2D and then do the whole pose estimation in 3D world space. This would allow us to make much better use of the kinematic constraints and it would also result in one joint pose estimate that can be back projected to each view instead of multiple (possibly different) pose estimates in the different views. In terms of testing it would be very interesting to see how the method performs when integrating more than two views.

References

[1] PETS 2009 Benchmark Data by University of Reading. http://www.cvg.rdg.ac.uk/PETS2009/a.html.

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial Structures Revisited: People Detection and Articulated Pose Estimation. In *CVPR*, 2009.

[3] Daniel Chen, Pi-Chi Chou, Clinton B. Fookes, and Sridha Sridharan. Multi-view Human Pose Estimation using Modified Five-point Skeleton Model. In *Int. Conference on Signal Processing and Communication Systems*, Dec. 2007.

[4] Stefano Corazza, Lars Mündermann, Emiliano Gambaretto, Giancarlo Ferrigno, and Thomas Andriacchi. Markerless Motion Capture through Visual Hull, Ar-

ticulated ICP and Subject Specific Model Generation. *International Journal of Computer Vision*, 87, 2010.

[5] E de Aguiar, C. Theobalt, M. Magnor, and H-P. Seidel. Reconstructing Human Shape and Motion from Multi-view Video. In *European Conference on Visual Media Production*, Nov. 2005.

[6] M. Eichner and V. Ferrari. We Are Family: Joint Pose Estimation of Multiple Persons. In *ECCV*, September 2010.

[7] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial Structures for Object Recognition. *International Journal of Computer Vision*, 61(1), Jan 2005.

[8] Vittorio Ferrari, Manuel J. Marn-Jimnez, and Andrew Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, 2008.

[9] P. Fihl and T.B. Moeslund. Pose Estimation of Interacting People using Pictorial Structures. In *Advanced Video and Signal Based Surveillance*, September 2010.

[10] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua. Multicamera People Tracking with a Probabilistic Occupancy Map. *PAMI*, 30(2), February 2008.

[11] N. GkalelIis, Hansung Kim, A. Hilton, N. Nikolaidis, and I. Pitas. The i3DPost Multi-View and 3D Human Action/Interaction Database. In *Conference for Visual Media Production*, Nov. 2009.

[12] Laetitia Gond, Patrick Sayd, Thierry Chateau, and Michel Dhome. A 3D Shape Descriptor for Human Pose Recovery. In *Articulated Motion and Deformable Objects*, volume 5098 of *LNCS*. Springer Berlin / Heidelberg, 2008.

[13] A. Gupta, A. Mittal, and L.S. Davis. Constraint Integration for Efficient Multiview Pose Estimation with Self-Occlusions. *PAMI*, 30(3), March 2008.

[14] M. Hofmann and D.M. Gavrila. Multi-view 3D Human Pose Estimation Combining Single-frame Recovery, Temporal Integration and Model Adaptation. In *CVPR*, June 2009.

[15] H. Jiang. Human Pose Estimation Using Consistent Max-Covering. In *ICCV*, 2009.

[16] Yusuke Matsumoto, Toshikazu Wada, Shuichi Nishio, Takehiro Miyashita, and Norihiro Hagita. Scalable and Robust Multi-people Head Tracking by Combining Distributed Multiple Sensors. *Intelligent Service Robotics*, 3(1), 2010.

[17] J. R. Mitchelson and A. Hilton. Simultaneous Pose Estimation of Multiple People using Multiple-View Cues with Hierarchical Sampling. In *British Machine Vision Conference*, 2003.

[18] D. Ramanan, D. A. Forsyth, and A. Zisserman. Tracking People by Learning their Appearance. *PAMI*, 29(1), Jan 2007.

[19] Leonid Sigal and Michael J. Black. Measure Locally, Reason Globally: Occlusion-sensitive Articulated Pose Estimation. In *CVPR*, 2006.

[20] Yang Wang and Greg Mori. Multiple Tree Models for Occlusion and Spatial Constraints in Human Pose Estimation. In *ECCV*, 2008.

[21] Jian Yao and Jean-Marc Odobez. Multi-Camera Multi-Person 3D Space Tracking with MCMC in Surveillance Scenarios . In *ECCV workshop on Multi Camera and Multi-modal Sensor Fusion Algorithms and Applications*, Oct. 2008.

Chapter 8

Conclusion

This thesis has presented methods to automate the analysis of human motion. Such automatic analysis can provide valuable information in many applications within very different areas like clinical diagnostics, video surveillance, and the computer game industry. The thesis has presented work within four overall topics, namely foreground segmentation, gesture recognition, gait type analysis, and human pose estimation.

Within foreground segmentation the thesis has presented a robust background subtraction method that allows for segmentation over long periods of time. The thesis has also shown how the most important parameters of the background subtraction method can be adjusted online to increase segmentation quality over time in dynamic environments.

Motion primitives were introduced for recognition of arm gestures. Both a 2D and a view-invariant 3D version of the motion primitives were presented, both showing good recognition rates also when it was not known when the execution of the gesture would commence.

Within gait type analysis a robust method was presented to classify the three main gait types, *i.e.* walking, jogging, and running. The method achieved recognition results comparable to state-of-the-art methods with far less training data and more diverse test data. The duty-factor was furthermore introduced to describe the gait types as one continuum instead of the sometimes ambiguous notion of three distinct classes.

For human pose estimation two methods were presented with the main focus on occlusion handling for interacting people. A single-view method was presented that optimized the combination of edge and appearance information for detection of body parts and furthermore handled occlusions without having to explicitly model the dependencies between body parts of multiple people. A multi-view extension of the pose estimation method was also presented. This method integrated body part probabilities from multiple cameras to improve the 2D pose estimates in each view. This made pose estimation possible in all views even though a person was heavily occluded in one view.

8.1 Discussion and outlook

The presented background subtraction method is capable of providing good quality foreground segmentation for long periods of time by updating the background model. It has unfortunately not been possible to make the ten hour long video sequence available to the research community due to privacy issues. The background subtraction model is relying solely on color and intensity information at pixel level and it will therefore always be sensitive to foreground with an appearance that match the background. The relatively low image quality provided by many cameras in real-life scenes significantly increases the noise in foreground masks from pixel-based methods. To address this issue other low level information can be incorporated in the background model, for example edge information, shadow detection, and pixel-wise temporal information. Region based information can also be used to improve foreground segmentation and lastly, high level information from a cognitive system can be fed back to the segmentation process to include contextual information.

Including more information in the segmentation method often leads to an increased processing load which in many cases calls for a careful balance between segmentation quality and processing time, especially when foreground segmentation is used as a preprocessing step. To address this problem foreground segmentation methods that can be executed as many parallel processes have been proposed with efficient implementations on graphics processing units.

The work presented on action recognition deals with a specific set of actions. The motion primitive is a principled approach that would work for many different actions but the simple extraction and compact representation of the primitives could cause the recognition rates to decrease if many actions are to be recognized by the same system. The gait type classification also targets a specific set of actions. Despite its very limited training data (a single execution of a stride for each gait type and only 270 frames in total) the recognition rates are comparable to state of the art action recognition methods. Although these methods often recognize a slightly larger set of actions they also use significantly larger training sets.

Recognizing complex actions and activities outside the laboratories is a challenge to state of the art methods and requires huge amounts of annotated training data to get a recognition that generalize well to different people. By using contextual information it could be possible to identify a small set of actions that is to be recognized in a given situation and the methods presented in this thesis could be applied in such a system.

The pose estimation methods presented in the thesis are partly motivated by the fact that they can facilitate the recognition of a wide range of different actions. The occlusion handling is seen as an important problem to address since occlusions are frequent in even slightly crowded scenes and since many interesting actions and activities involve multiple people. When using appearance based methods for tracking and pose estimation of multiple people one should note that people tend to wear clothes with similar appearances (*e.g.* blue jeans, and brown or gray colored

jackets). Hence, appearance based methods most often gain from including other types of information.

Human pose estimation using a single camera has in recent years resulted in solid and successful systems. However, with the level of occlusions that arise from interacting people and crowded scenes it seems that multi-camera approaches need to be applied in most real-life scenarios. Developing systems that can efficiently handle more cameras and also cameras with higher video resolution and color quality will improve the applicability of human pose estimation significantly.

Appendix A

Data set listings

This appendix provides a comparative listing of data sets relevant to recognition of human actions and activities. The appendix also lists a few data sets for tracking and pose estimation and finally references to other listings of data sets within the areas of human motion analysis and computer vision in general.

Table A.1 lists data sets concerned with action recognition, table A.2 lists multi-view data sets for activities and interactions, and table A.3 list single-view data sets for activities. The tables do not provide a complete listing of all data sets available but list the characteristics of the most relevant data sets for the overall topic of this thesis.

Pose estimation and tracking

All of the following data sets are publicly available.

PETS 2009 Tracking data set. PETS2009 and other versions of the PETS data sets have had great impact on evaluation of tracking methods.

- Website: www.cvg.rdg.ac.uk/PETS2009/

EPFL CVLab multi-view data set Tracking data set. Videos of multiple people from three to four cameras. Camera calibrations are available for some sequences.

- Website: cvlab.epfl.ch/data/pom/

Multiple Person Tracking Data sets Tracking data set. Two videos of four and five people walking in an office space. 15 cameras are used.

- Website: www.cs.cmu.edu/∼abhinavg/Downloads.html

ETHZ PASCAL Stickmen Pose estimation data set. Provides manually annotated stick figures on still images. Also the Buffy Stickmen data set is available from the same website.

- Website: www.vision.ee.ethz.ch/∼calvin/datasets.html

Multi-view 3D Human Pose Pose estimation data set. A three-camera data set of two people walking, gesturing, and interacting.

- Website: www.gavrila.net/Research/3-D_Human_Body_Tracking/CVPR_2009_data_set/cvpr_2009_data_set.html

CVMT pose Pose estimation data set. Single-view video of interacting people. Ground truth is available for sample frames.

- Website: www.cvmt.dk/∼pfa/pose/index.html

Other data set listings

CANTATA data set listing Extensive listing of data sets within video surveillance, medical images, and consumer applications.

- Website: www.hitech-projects.com/euprojects/cantata/datasets_cantata/dataset.html

VisionBib data sets listing Extensive list with many different topics.

- Website: datasets.visionbib.com/info-index.html

Computer Vision Online Also an extensive list covering broadly the area of computer vision

- Website: computervisiononline.com/datasetslist

Table A.1: Action data sets.

Name	HERMES gesture - CVMT	KTH Action recognition	Weizmann action recognition	CVMT Gait	UT-Tower data set	HumanEva
Type	Actions	Actions	Actions	Actions	Actions	Actions
Availability	Public	Public	Public	Public	Public	Registration required
Website	[1]	[2]	[3]	[4]	[5]	[6]
Ground truth data	Action label per sequence	Action labels per sequence	Action label per sequence	Action labels	Action labels	Action labels and motion capture data
Video resolution, grayscale/color	640x480 [g]	160x120 [g]	180 x 144 [c]	640x480 [c]	360 x 240 [c]	640x480 [4 g , 3 c]
#sequences x #frames x fps	66 x 900 x 25	600 x 50 x 25	90 x 50 x 50	10 x 1000 x 30	2 x x 10	54 x 400 x 60
#people (height in px) x #repetitions	11 ($\tilde{3}$00) x 10	25 ($\tilde{1}$00) x 16	9 ($\tilde{7}$0) x 10	1 ($\tilde{1}$80) x 5-10	6 ($\tilde{2}$0) x 2	4 ($\hat{2}$75) x 1-3
#cameras	1	1	1	1	1	7
Actions performed	Move closer, move left, move right, point forward, point right, raise arm	Box, clap, wave, run, jog, walk	Run, walk, skip, jumping-jack, jump-forward-on-two-legs, jump-in-place-on-two-legs, gallop sideways, wave two-hands, wave one-hand, bend	Slow walk, normal walk, fast walk, jogging, running, and sprinting	Pointing, standing, digging, walking, carrying, running, wave one arm, wave two arm, jumping	Walking, jogging, throw/catch, gesture ("hello", "goodbye"), box, combo (walking, jogging, and balancing on each foot performed without interruption)
Related publication	[19]	[26]	[17]	[20]	[18]	[27]
Indoor / outdoor	Indoor	Both	Outdoor	Indoor	Outdoor	Indoor
Lighting conditions	Diffuse light	Diffuse light and sunshine	Diffuse	Diffuse	Direct sunlight	Diffuse
Scene characteristics and simplifications	Stationary background, frontal view	Stationary background	Stationary simple background	Stationary simple background	Simple background. Slightly jittering camera	Stationary simple background
Camera calibration	No	No	No	No	No	Yes
Observations / experience		Not completely static camera, poor image quality at times. 4 different scenarios recorded: Indoor, side view, diagonal view, changed clothes	Robustness videos also available - occlusions and such. Background subtraction results available for some sequences	The direction of motion changes in some sequences. Empty background frames are provided	Bounding boxes and foreground segmentation available	Synchronized cameras. Background subtraction results available. Mesh model available for one subject. Baseline implementation available.

Table A.2: Multi-view activity data sets.

Name	i3DPost Multi-view Human Action Data sets	MuHAVi: Multi-camera Human Action Video Data	The TUM Kitchen Data Set	BEHAVE Multiagent Interaction Data set	UCR-Videoweb	IXMAS data set
Type	Actions and activities	Actions and activities	Activities	Activities	Activities and interactions	Activities
Availability	Registration required	Registration required	Public	Public	Registration required	Public
Website	[7]	[8]	[9]	[10]	[11]	[12]
Ground truth data	Action labels, motion capture	Action labels per sequence	Activity labels, motion capture data	Activity labels specified by frame numbers	Activity labels specified by frame numbers	Activity labels per frame
Video resolution, grayscale/color	HD video [c]	720 x 576 [c]	384x288 or 780x582 [c]	640x480 [c]	640x480 [c]	390x291 [c]
Size of data set	104 executions in total by 8 people	119 executions in total by 7 people	21 sequences of 4 people	4 long clips with 2-5 people interacting	51 sequences each with multiple people	429 sequences in total by 11 people
#cameras	8	8	4	2	4-8	5
Actions performed	Walk, Run, Jump, Bend, Hand-wave, Jump-in-place, Sit-StandUp, Run-fall, Walk-sit, Run-jump-walk, Handshake, Pull, Facial-expressions	Walk-TurnBack, Run-Stop, Punch, Kick, ShotGun-Collapse, Pull-Heavy-Object, Pickup-ThrowObject, Walk-Fall, Look-In-Car, Crawl-On-Knees, Wave-Arms, Draw-Graffiti, JumpOverFence, DrunkWalk, Climb-Ladder, Smash-Object, Jump-Over-Gap	Setting a table, reaching and grasping	InGroup, approach, walk together, meet, split, ignore, chase, fight, run together, following	Throwing a ball, shaking hands, standing in a line, handing out forms, running, limping, getting into/out of a car, cars making turns, and more	Check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick, point, pick up, throw (over head), throw (from bottom up)
Related paper	[21]	[28]	[30]	-	[29]	[31]
Indoor / outdoor	Indoor studio	Indoor studio	Indoor	Outdoor	Outdoor	Indoor
Lighting conditions	Diffuse	Diffuse but uneven	Diffuse + sunlight from windows	Cloudy	Direct sunlight and shadow	Diffuse
Scene characteristics and simplifications	All-blue background	Stationary complex background	Setting the table one item at the time (mostly)	Complex background, almost static	Rather complex backgrounds	Stationary simple background
Camera calibration	Yes	No	Yes	Corresponding pixel positions	No	Yes
Observations / experience	Background images, and 3D mesh models available	Camera calibration can be calculated. Virtual actors also available	Data from RFID tags and magnetic sensors available	Some ground truth bounding boxes available		Reconstructed volumes available in Matlab format

Table A.3: Single-view activity data sets.

Name	Hollywood Human Actions I	Hollywood Human Actions II	YouTube Action Data set	UT-Interaction data set
Type	Activities	Activities	Activities	Activities and interactions
Availability	Public	Public	Public	Public
Website	[13]	[14]	[15]	[16]
Ground truth data	Activity labels specified by frame numbers	Activity labels per sequence	Activity labels per sequence	Activity labels specified with time intervals and bounding boxes
Video resolution, grayscale/color	Varying x 240 [g+c]	Varying x 240 [g+c]	Youtube videos	720x480 [c]
Size of data set	+600 action samples from 32 movies	+3600 movie clips	+1000 sequences	20 sequences of 15 people
#cameras	1	1	1	1
Actions performed	Answer phone, get out of car, hand shake, hug person, kiss, sit down, sit up, stand up	AnswerPhone, DriveCar, Eat, FightPerson, GetOutCar, HandShake, HugPerson, Kiss, Run, SitDown, SitUp, StandUp	Basketball shooting, biking/cycling, diving, golf swinging, horse back riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog	Shake-hands, point, hug, push, kick and punch
Related publication	[22]	[24]	[23]	[25]
Indoor / outdoor	Both (from movies)	Both (from movies)	Both	Outdoor
Lighting conditions	Varying	Varying	Varying	Diffuse
Scene characteristics and simplifications	Complex scenes	Complex scenes	Varying	Two backgrounds. One stationary simple background, one with little motion
Camera calibration	No	No	No	No
Observations / experience		Additional scene classes available		

References

[1] www.cvmt.dk/projects/Hermes/gesture-data.html.

[2] www.nada.kth.se/cvap/actions/.

[3] www.wisdom.weizmann.ac.il/∼vision/SpaceTimeActions.html#Database.

[4] www.cvmt.dk/projects/Hermes/gait-data.html.

[5] cvrc.ece.utexas.edu/SDHA2010/Aerial_View_Activity.html.

[6] vision.cs.brown.edu/humaneva/.

[7] kahlan.eps.surrey.ac.uk/i3dpost_action/.

[8] dipersec.king.ac.uk/MuHAVi-MAS/.

[9] oldias.informatik.tu-muenchen.de/download/kitchen-activity-data.

[10] groups.inf.ed.ac.uk/vision/BEHAVEDATA/INTERACTIONS/index.html.

[11] cvrc.ece.utexas.edu/SDHA2010/Wide_Area_Activity.html.

[12] 4drepository.inrialpes.fr/public/viewgroup/6.

[13] www.irisa.fr/vista/actions/.

[14] www.irisa.fr/vista/actions/hollywood2/.

[15] server.cs.ucf.edu/∼vision/projects/liujg/YouTube_Action_dataset.html.

[16] cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html.

[17] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as Space-Time Shapes. In *ICCV*, Washington, DC, USA, 2005.

[18] Chia-Chih Chen and J.K. Aggarwal. Recognizing Human Action From a Far Field of View. In *Workshop on Motion and Video Computing*, Dec. 2009.

[19] P. Fihl, M.B. Holte, and T.B. Moeslund. Motion Primitives and Probabilistic Edit Distance for Action Recognition. In *Gesture-Based Human-Computer Interaction and Simulation, LNCS, vol. 5085, Springer Berlin/Heidelberg*, January 2009.

[20] P. Fihl and T.B. Moeslund. Invariant Gait Continuum Based on the Duty-factor. *Signal, Image and Video Processing*, 3, 2009.

[21] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, and I. Pitas. The i3DPost Multi-view and 3D Human Action/interaction. In *Conference for Visual Media Production*, 2009.

[22] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, Benjamin Rozenfeld, Inria Rennes, Irisa Inria Grenoble, and Lear Ljk. Learning Realistic Human Actions from Movies. In *CVPR*, 2008.

[23] Jingen Liu, Jiebo Luo, and M. Shah. Recognizing Realistic Actions from Videos "In the Wild". In *CVPR*, June 2009.

[24] Marcin Marszałek, Ivan Laptev, and Cordelia Schmid. Actions in Context. In *CVPR*, 2009.

[25] M.S. Ryoo and J.K. Aggarwal. Spatio-temporal Relationship Match: Video Structure Comparison for Recognition of Complex Human Activities. In *ICCV*, Oct 2009.

[26] C. Schüldt, I. Laptev, and B. Caputo. Recognizing Human Actions: a Local SVM Approach. In *International Conference on Pattern Recognition*, 2004.

[27] L. Sigal and M. J. Black. HumanEva: Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human Motion. In *Techniacl Report CS-06-08, Brown University*, Dec. 2006.

[28] S Singh, S.A. Velastin, and H Ragheb. MuHAVi: A Multicamera Human Action Video Dataset for the Evaluation of Action Recognition Methods. In *Workshop on Activity monitoring by multi-camera surveillance systems*, August 2010.

[29] Bi Song, A.T. Kamal, C. Soto, Chong Ding, J.A. Farrell, and A.K. Roy-Chowdhury. Tracking and Activity Recognition Through Consensus in Distributed Camera Networks. *IEEE Transactions on Image Processing*, 19(10), oct. 2010.

[30] M. Tenorth, J. Bandouch, and M. Beetz. The TUM Kitchen Data Set of Everyday Manipulation Activities for Motion Tracking and Action Recognition. In *International Conference on Computer Vision Workshops*, Oct. 2009.

[31] Weinland, D., E. Boyer, and R. Ronfard. Action Recognition from Arbitrary Views using 3D Exemplars. In *ICCV*, Oct. 2007.

Media Technology
Aalborg University
Niels Jernes Vej 14
DK-9220 Aalborg
Denmark

Telephone: +45 9940 8793
TeleFax: +45 9940 9788

URL: http://www.create.aau.dk

ISBN 978-87-992732-4-9