Exploring New Waters

Advancing Fish Monitoring with Computer Vision

Pedersen, Malte

DOI (link to publication from Publisher):
10.54337/aau548864167

Publication date:
2023

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

EXPLORING NEW WATERS: ADVANCING FISH MONITORING WITH COMPUTER VISION

BY
MALTE PEDERSEN

DISSERTATION SUBMITTED 2023

AALBORG UNIVERSITY
DENMARK

Exploring New Waters: Advancing Fish Monitoring with Computer Vision

Ph.D. Dissertation

Malte Pedersen

Dissertation submitted June 9, 2023

Dissertation submitted: June 9, 2023

PhD supervisor: Prof. Thomas B. Moeslund
Aalborg University

PhD committee: Associate Professor Georgios Triantafyllidis (chair)
Aalborg University, Denmark

Professor Morten Goodwin
University of Agder, Norway

Professor Serge Belongie
University of Copenhagen, Denmark

PhD Series: Technical Faculty of IT and Design, Aalborg University

Department: Department of Architecture, Design
and Media Technology

ISSN (online): 2446-1628
ISBN (online): 978-87-7573-693-5

Published by:
Aalborg University Press
Kroghstræde 3
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

Printed in Denmark by Stibo Complete, 2023

Curriculum Vitae

Malte Pedersen

Malte Pedersen received his bachelor's degree in computer science in 2015 and master's degree with specialization in computer vision in 2017, both from Aalborg University in Denmark. While studying for his master's degree he spent a semester as an intern at SINTEF in Oslo assisting in the development of an underwater range-gated time-of-flight camera named UTOFIA. His master's thesis focused on tracking multiple zebrafish in an aquarium environment using a custom stereo vision setup and the work led to publications at CVPR.

He was employed as a research assistant at the Visual Analysis and Perception laboratory (VAP lab) at the Department of Architecture, Design, and Media Technology at Aalborg University from 2017-2020 prior to becoming a PhD student. During this time he supervised or co-supervised more than 15 undergraduate and graduate projects and was, among other things, a teacher's assistant on the 'Programming of Complex Software Systems' course.

His research has mainly been focused on the underwater domain and he developed the 3D multiple object tracking benchmark dataset (3D-ZeF) and an associated tracking pipeline in collaboration with Stefan H. Bengtson and Joakim B. Haurum. Furthermore, he designed and constructed an underwater camera and lighting system that was used to capture data from Limfjorden, which laid the foundation for a popular marine object detection dataset named the Brackish dataset. Concurrently, he was part of multiple projects in other domains including monitoring of public sports fields using thermal cameras, detecting people in the Copenhagen Metro using thermal cameras, water level estimation in sewers based on RGB images, predicting the state of spray nozzles from RGB images, and more.

Curriculum Vitae

Abstract

This thesis focuses on advancing the use of computer vision in underwater environments, specifically in the areas of multi-object tracking, dataset development, and re-identification. The work revolves around three aspects of underwater research, namely: behavioral analysis of fish in a controlled aquarium setup, detection and tracking of marine organisms in the wild, and long-term monitoring of fish in the wild based on re-identification.

The first part of the thesis focuses on stereo camera calibration and tracking of multiple fish in an aquarium. An objective comparison of three common 3D reconstruction procedures was conducted, evaluating their flexibility, ease of use, and precision. Our findings revealed that the optimal approach consisted of an independent calibration of the intrinsic and extrinsic camera parameters while using ray tracing and Snell's law to account for refraction during reconstruction. We actively used this procedure to accurately reconstruct the 3D positions of zebrafish as part of a novel 3D multi-object tracker which we evaluated on the first ever publicly available underwater 3D multi-object tracking benchmark dataset. Additionally, we proposed an occlusion complexity estimation metric as an alternative to the conventional and misleading 'number of objects' to describe the difficulty of tracking multiple zebrafish. Building upon the occlusion complexity estimation metric, further advancements were made in evaluating multi-object tracking sequences in a broader sense. This resulted in the development of the objective and accurate multi-object tracking dataset complexity metric named MOTCOM.

In the second part of the thesis the research revolves around an underwater camera system mounted in a local Danish strait to record marine organisms in the wild. This led to the creation of a publicly available bounding box annotated underwater object detection dataset captured in an environment with varying visibility. This was later on expanded with additional sequences and multi-object tracking annotations. State-of-the-art detectors and trackers were fine-tuned and evaluated on the datasets to provide baselines to promote further research in the field. Additionally, we proposed a framework for generating realistic synthetic sequences of multiple fish exhibiting social group behavior and investigated ways of using synthetic data for training a tracker.

Lastly, based on the experience gained from developing our datasets we presented critical factors to consider for optimizing data collection in challenging underwater environments.

In the context of re-identification, a pipeline based on keypoint matching was proposed for identifying giant sunfish. The main advantage of the solution is that it consists of common algorithms configured with default settings, meaning that it requires no training or parameter fine-tuning. We found that a state-of-the-art deep learning based keypoint descriptor outperformed traditional handcrafted features descriptors with respect to the re-identification task. To enhance the performance of the pipeline, we introduced a contrast enhancement module. It not only offered practical utility but also improved the performance of the handcrafted feature descriptors. Furthermore, a solution for easing manual verification through image alignment based on homography estimation was presented. Lastly, the limitations of ranked list outputs in re-identification systems were discussed, and a binary classification approach was suggested, effectively reducing the number of false positives.

Together, this thesis contributes to the advancement of computer vision in underwater environments, providing valuable insights, benchmark datasets, metrics, and practical solutions for a variety of applications in the field.

Resumé

Denne afhandling fokuserer på at fremme anvendelsen af computer vision i undervandsmiljøer, specifikt inden for områderne multi-objekt tracking, udvikling af datasæt, og re-identifikation. Arbejdet handler hovedsageligt om automatisering af tre aspekter af undervandsmonitorering: adfærdsanalyse af fisk i et kontrolleret miljø, monitorering af marine organismer i deres naturlige omgivelser, og langvarig overvågning af vilde fisk.

Den første del af afhandlingen fokuserer på kalibrering af et stereo-kamera system og 3D tracking af fisk i et akvarium. Vi foretog en objektiv sammenligning af tre typiske 3D-rekonstruktionsmetoder, hvor de blev vurderet på deres fleksibilitet, brugervenlighed, og præcision. Resultaterne viste, at den optimale tilgang indebærer en separat kalibrering af kameraernes intrinsiske og ekstrinsiske parametre, samtidig med at der tages højde for refraktion ved hjælp af ray-tracing og Snells lov under rekonstruktionsdelen. Denne procedure blev aktivt brugt til at rekonstruere 3D-positioner af zebrafisk som en del af vores 3D multi-objekt tracker, som vi evaluerede på det første undervands 3D multi-objekt tracking benchmark dataset som vi publicerede sammen med trackeren. Derudover foreslog vi en metode til at estimere sværhedsgraden af okklusion i multi-objekt tracking sekvenser som et alternativ til den konventionelle og misvisende 'antal objekter', der typisk bruges til at beskrive sværhedsgraden af at tracke flere objekter. Gennem yderligere arbejde på estimering af kompleksiteten i multi-objekt tracking sekvenser lykkedes det os at udvikle tre objektive metrikker til at beskrive sværhedsgraden af okklusion, visuelle ligheder mellem objekter, og objekternes bevægelsesmønstre. Dette resulterede i det første objektive og præcise kompleksitetsmål til at beskrive sværhedsgraden af multi-objekt tracking datasæt, kaldet MOTCOM.

I anden del af afhandlingen er omdrejningspunktet et undervandskamerasystem, der har været monteret i Limfjorden i en længere periode for at optage marine organismer i deres naturlige miljø. Dette ledte til udviklingen af et offentligt tilgængeligt datasæt med bounding box annoteringer af marine organismer optaget i varierende sigtbarhed. Dette datasæt blev senere udvidet med yderligere sekvenser og multi-objekt tracking annoteringer. Vi fintunede og evaluerede nogle af de nyeste detektorer og trackere på disse

datasæt for at præsentere et udgangspunkt for yderligere forskning inden for feltet. Derudover præsenterede vi et framework til generering af realistiske syntetiske sekvenser af fisk, der udviser social gruppeadfærd, og vi undersøgte måder at bruge syntetisk data til træning af en tracker. Endelig, baseret på den erfaring vi har fået gennem udviklingen af vores datasæt præsenterede vi et sæt af kritiske faktorer, der bør tages højde for, for at optimere dataindsamling i undervandsmiljøer.

I den sidste del af afhandlingen præsenteres en pipeline for re-identifikation af klumpfisk baseret på keypoint matching. En af fordelene ved vores løsning er, at den består af nemt tilgængelige algoritmer konfigureret med standardindstillinger, hvilket betyder, at den kræver hverken træning eller finjustering af parametre. Vi kom frem til at en deep learning-baseret keypoint deskriptor præsterede bedre end traditionelle håndfremstillede deskriptorer i forhold til re-identifikationsopgaven. For at forbedre ydeevnen af vores pipeline introducerede vi et kontrastforstærkningsmodul. Det betød ikke blot en forbedring til den praktiske anvendelse af algoritmen, men forbedrede også resultaterne af de håndfremstillede deskriptorer. Derudover præsenterede vi en løsning til at gøre den manuelle verifikation lettere ved hjælp af en projektering af billederne baseret på homografiestimering. Til sidst, blev begrænsningerne ved rangerede lister som output i re-identifikationssystemer diskuteret, og der blev foreslået at anskue problemet som værende binær klassifikation udfra om to billeder viser samme individ eller ej, hvilket effektivt reducerer antallet af falske positiver.

Samlet set bidrager denne afhandling til fremskridt inden for computer vision i undervandsmiljøer ved at levere værdifulde indsigter, benchmarkdatasæt, metrikker, og praktiske løsninger.

Contents

Thesis Details

Thesis Title:	Exploring New Waters: Advancing Fish Monitoring with Computer Vision
PhD Student:	Malte Pedersen
Supervisor:	Prof. Thomas B. Moeslund, Aalborg University

The thesis consists of the following publications:

[A] ***Malte Pedersen***, Stefan H. Bengtson, Rikke Gade, Niels Madsen, and Thomas B. Moeslund, "Camera Calibration for Underwater 3D Reconstruction Based on Ray Tracing using Snell's Law". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1410-1417, 2018.

[B] ***Malte Pedersen***, Joakim B. Haurum, Stefan H. Bengtson, and Thomas B. Moeslund, "3D-ZeF: A 3D Zebrafish Tracking Benchmark Dataset". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2426-2436, 2020.

[C] ***Malte Pedersen***, Joakim B. Haurum, Patrick Dendorfer, and Thomas B. Moeslund, "MOTCOM: The Multi-Object Tracking Dataset Complexity Metric". In: *Computer Vision – ECCV 2022*, Lecture Notes in Computer Science, vol 13668, pp. 20–37, 2022.

[D] ***Malte Pedersen***, Niels Madsen, and Thomas B. Moeslund, "No Machine Learning Without Data: Critical Factors to Consider when Collecting Video Data in Marine Environments". In: *The Journal of Ocean Technology*, vol 16 (3), pp. 21-30, *essay*, 2021.

[E] ***Malte Pedersen***, Joakim B. Haurum, Rikke Gade, Niels Madsen, and Thomas B. Moeslund, "Detection of Marine Animals in a New Underwater Dataset with Varying Visibility". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 18-26, 2019.

[F] ***Malte Pedersen***, Daniel Leyhotský, Ivan Nikolov, and Thomas B. Moeslund, "BrackishMOT: The Brackish Multi-Object Tracking Dataset". In: *Proceedings of the 23rd Scandinavian Conference on Image Analysis (SCIA)*, Part I, pp. 17-33, 2023.

[G] ***Malte Pedersen***, Joakim B. Haurum, Thomas B. Moeslund, and Marianne Nyegaard, "Re-Identification of Giant Sunfish using Keypoint Matching". In: *Proceedings of the Northern Lights Deep Learning Workshop*, vol 3, 2022.

[H] ***Malte Pedersen***, Marianne Nyegaard, and Thomas B. Moeslund, "Finding Nemo's Giant Cousin: Keypoint Matching for Robust Re-identification of Giant Sunfish". In: *Journal of Marine Science and Engineering (JMSE)*, vol 11, 2023.

In addition to the aforementioned, the PhD student has co-authored the following publications which are not part of the thesis.

- Niels Madsen, ***Malte Pedersen***, Kurt T. Jensen, Peter R. Møller, Rasmus E. Andersen, and Thomas B. Moeslund, "Fishing with C-TUCs (Cheap Tiny Underwater Cameras) in a Sea of Possibilities". In: *The Journal of Ocean Technology*, vol 16 (2), pp. 19-30, *essay*, 2021.
- Joakim B. Haurum, Chris H. Bahnsen, ***Malte Pedersen***, and Thomas B. Moeslund, "Water level estimation in sewer pipes using deep convolutional neural networks". In: *Water*, vol 12, 2020.
- Joakim B. Haurum, Anastasija Karpova, ***Malte Pedersen***, Stefan H. Bengtson, and Thomas B Moeslund, "Re-identification of zebrafish using metric learning". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pp. 1-11, 2020.
- Jakob S. Lauridsen, Julius A. G. Graasmé, ***Malte Pedersen***, David G. Jensen, Søren H. Andersen, and Thomas B. Moeslund, "Reading Circular Analogue Gauges using Digital Image Processing". In: *14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, pp. 373-382, 2019.
- Anders Jørgensen, ***Malte Pedersen***, Rikke Gade, Jens Fagertun, and Thomas B. Moeslund, "Reaching Behind Specular Highlights by Registration of Two Images of Broiler Viscera". In: *Computer Vision – ACCV 2018 Workshops*, Lecture Notes in Computer Science, vol 11367, 2018.
- Mathias Z. Vestergaard, Stefan H. Bengtson, ***Malte Pedersen***, Christian Rankl, and Thomas B. Moeslund, "Stitching rid-wise Atomic Force Microscope Images". In: *11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, pp. 110-117, 2016.

Preface

This dissertation is organized as a compendium of papers aimed at partially fulfilling the requirements for the attainment of a PhD degree at the Section of Media Technology, Aalborg University, Denmark. I would like to express my deep gratitude to the Independent Research Fund Denmark (DFF) for providing funding for this thesis under the case number 9131-00128B. The central theme of the work is an exploration of methods and development of underwater datasets for improving fish monitoring through the application of computer vision. The thesis consists of an introductory part that provides an overview of the research, including three chapters that describe the work within tracking fish in a controlled environment, detection and tracking marine organisms in the wild, and long-term tracking of individual giant sunfish. This is followed by a second part containing the papers laying the foundation for the thesis. However, before we dive into this underwater adventure, I would like to take a moment to express my heartfelt gratitude to a select few who have played a significant role in this journey.

If there is a recipe for a typical PhD student, I am sure I did not follow it. If you asked me during or after high school I would have told you that I would study to become a paramedic; I was not interested in pursuing higher education. Three years later I signed up for a bachelor's program at Aalborg University. Maybe it was the harmonious title of the program: "Internet Technologies and Computer Systems" that lured me in, but I am not sure. I basically had no idea what I was getting myself into and I had never touched a programming language before. It became three life-defining years for me in several ways.

I am not sure if it was because of the aforementioned title of the program, but for some reason only a handful of students signed up, fifteen if I remember correct. After two years we were seven students left and it is important for me to thank them all. Had it not been for Anders Kalør, Morten Bonderup, Kristoffer Vestergaard, Lasse Bromose, Andreas Aakerberg, and Stefan Bengtson I would most likely have played computer games and dropped out of the program early on. Instead, you guys taught me to enjoy learning and give it all it takes to create some amazing projects.

During my bachelor's degree I worked on projects that included aspects of computer vision and image processing and it caught my attention. Therefore, I chose a specialization in informatics on my final semester. It was my first encounter with Thomas Moeslund who was both my teacher and supervisor that year. This was another defining moment for me. For those of you who do not know Thomas; he is the type of guy that magically makes whatever topic he is talking about interesting. Suddenly I had no doubt that computer vision was my future. Thank you Thomas, I am extremely grateful that you inspired me all those years back and continue to do so to this day.

Following my bachelor's degree I continued on a master's program in vision, graphics, and interactive systems. At this point my mind was focused on computer vision and everything else on the program that rhymed on graphics or interaction was a bummer. But once again, Morten, Andreas, and especially Stefan were pillars that guided me through ups and downs. A highlight during this period was also the therapeutic breaks from studying with fresh air, frisbee-golf, and long talks with fellow student Christoffer Rasmussen, which led to an appreciated long-lasting friendship.

In the end, all of this led to an employment as a research assistant at the visual analysis and perception lab headed by Thomas Moeslund. A position I was really fond of as I got the opportunity to work in various fields of computer vision and with a range of people. Especially my collaborations with Joakim Bruslund Haurum and Stefan Bengtson were extremely rewarding and a decisive factor for me to pursue the PhD degree. A huge thanks to both of you for enduring my endless questions and discussions. I must also express my deep gratitude to Professor Niels Madsen, who has been the visionary behind this entire underwater adventure, Dr. Marianne Nyegaard, with whom I have searched for the elusive giant sunfish, and Dr. Tom Trnski for the great coffee-talks and for hosting me at the Natural Sciences department at Auckland War Memorial Museum in New Zealand during my stay abroad.

However, most importantly of all, it was through my employment at the lab that I met the love of my life, Rikke Gade, with whom I now have the most beautiful, curious, and cheerful girl, Karla, and twins on the way. I cannot emphasize enough how much I appreciate the life we have together. I wake up happy every single day.

Last, but not least, none of this would have been possible without the most open-minded, warm-hearted, and supportive family that always have my back. I am extremely fortunate, thank you.

Malte Pedersen
Aalborg University, June 2023

Part I

Hold your Breath: An Overview of the Thesis

Chapter 1

Introduction

As the world's population continues to grow, the demand for sustainable food sources is increasing [21]. More than a third of the world's population is located at or near coastal areas and many people depend on fish as their primary source of protein [2]. Technological advancements in the fishing industry, such as more efficient and effective fishing gear, have made it easier to catch larger quantities of fish. The increase in fishing has led to concerns about overfishing, depletion of fish populations, and potential negative impacts on marine ecosystems across the globe [20]. Concerns for the impact of overfishing continues to grow and with it a rising demand for enhanced monitoring to enable effective regulation and ensure restoration and conservation of fish populations and ecosystems.

Apart from the growing population and demand for food, the world is also struggling with climate change, characterized by a rise in temperatures, pollution, and other related problems [20]. This has led to a need for enhanced monitoring and analysis of various environmental factors to better understand the implications of these changes on marine ecosystems. The urgency of this need has been brought to the forefront in recent years with the introduction of the United Nations' fourteenth Sustainable Development Goal (UN SDG #14) *Life Below Water* [19] and as the scale and impact of the climate crisis become increasingly apparent to people. As a result, there is a growing interest in developing new tools and technologies for monitoring and analyzing marine environments.

However, assessing the state of marine environments is a multifaceted process, which varies depending on the situation and may entail everything from sonar monitoring to dissection of organisms [12, 18]. In recent years, the utilization of computer vision has gained attention due to the emergence of cheaper, smaller, and better underwater cameras [11]. This method provides a non-intrusive solution for detecting and tracking marine organisms in their

natural habitats, as opposed to traditional techniques which often involve catching and potentially hurting the organisms. The purpose of this thesis is to explore and advance the field of computer vision in underwater environments by addressing the overall research question: What are the limitations and challenges associated with detecting and tracking fish in diverse underwater environments using computer vision, and how can they be overcome?

1 Data: The Oxygen of Machine Learning

The widespread availability and affordability of powerful hardware has led to a steep increase in popularization of computer vision and machine learning during the past decades. These technologies have transitioned from being mostly used in the industry and academia to becoming a pervasive part of our daily lives. Tasks that were once achievable only by trained professionals using sophisticated computer systems are now becoming routine and accessible to anyone with a smartphone or tablet.

A major driver for reaching this point was the development of the novel deep learning architecture behind AlexNet [9] in 2012. AlexNet exhibited a superior level of performance in image classification in comparison to other contemporary approaches and it is widely regarded as one of the catalysts for the deep learning era that followed and which has lead to significant improvements within nearly every sub-field of computer vision and machine learning. Another event a few years earlier that was potentially as significant, if not more so, was the creation of the ImageNet [4] dataset, consisting of more than a million labeled images, which provided the necessary large-scale training data required for training deep neural models like AlexNet. Together, these breakthroughs sparked a rapid development in deep learning research, and have since enabled remarkable progress in various other fields such as natural language processing, speech recognition, and robotics, to name a few.

It is essential to underscore that the unprecedented performance demonstrated by AlexNet for a large part owes its existence to ImageNet. Beyond catalyzing a substantial proliferation of deep learning research, ImageNet spurred a corresponding surge in the domain of data acquisition, labeling, and synthesizing, which has lead to the development of an increasing number of large labelled datasets like MS COCO [10], Places [23], KITTI [6], MegaFace [8], and more [1, 3, 5, 7, 22, 24]. The abundance of data has been a key factor in the maturity of fields like image classification, facial recognition, autonomous driving, and surveillance. However, the same rapid advancement has not been proven within less popular fields, like marine monitoring, where annotated data has not been readily accessible or easily acquired. The number of labelled underwater image datasets is generally scarce and the datasets are small compared to their terrestrial counterparts [15].

Only recently has the notion gained traction that the ocean's resources are not infinite and require careful conservation and nurturing, which calls for large-scaled and long-term monitoring. And although the majority of the Earth's surface is covered in water, and many people rely on it as a source of sustenance, it remains an extremely inhospitable environment for humans to navigate. Unlike the terrestrial domain where people can capture beautiful images and videos simply using their phone, acquisition of high quality underwater data requires specialized equipment and significant planning. Moreover, annotating marine data is often a complex process, as underwater images are typically subject to turbidity, sub-optimal lighting, and challenging scenes. In addition to these technical challenges, the unpredictable nature of marine environments can make it a difficult and time-consuming task to capture images of interest. As a result, the development of comprehensive and diverse underwater image datasets remains an ongoing challenge.

2 Charting the Course: Thesis Structure

A focus of this thesis is development and utilization of computer vision algorithms for monitoring marine life in the wild. However, jumping straight into deep water without proper preparation is not advisable. As a precautionary measure, we adopt a step-by-step approach by starting with a controlled aquarium environment to gain knowledge. Next, we take a dive into shallow local waters to monitor marine life in the wild, and, finally, we extend our scope to deep international waters with long-term monitoring of fish. The limitation in underwater datasets is evident in this thesis as suitable datasets were not publicly available; instead, new datasets were designed, acquired, and annotated during the course of the research.

As briefly mentioned, the thesis is structured into three chapters, each exploring distinct aspects of underwater monitoring. Each chapter contains related work, which is integrated within the body text to maintain an engaging reading experience, and concludes with its own list of references.

The first part, presented in Chapter 2, centers on behavioral analysis of fish through 3D multi-object tracking in a controlled environment. The second part, presented in Chapter 3, delves into detection and tracking of multiple marine organisms in the wild. In the final part, presented in Chapter 4, the focus is on long-term monitoring of giant sunfish using re-identification. An overview of the chapters and the associated papers is presented in Figure 1.1. The arrows in the figure indicate direct influence from previous work, e.g., the findings in Paper A laid the foundation for parts of the work in Paper B; this will be elaborated in the respective chapters. A brief introduction to each of the papers is presented below.

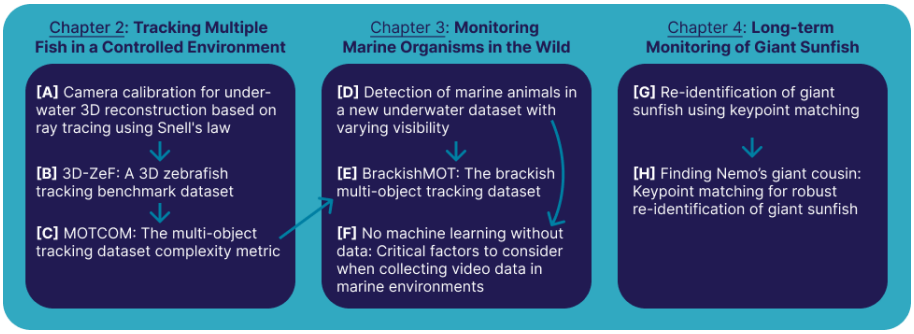


Fig. 1.1: The three boxes contain the titles of the papers that lay the foundation for the respective chapter indicated by the title above the box. The arrows illustrate direct influence by prior work.

Chapter 2: Tracking Multiple Fish in a Controlled Environment

We conducted three related studies on 3D multi-object tracking (MOT) of fish. In Paper A, we identified the three most commonly used methods for calibrating stereo cameras for underwater 3D reconstruction and conducted the first objective comparison between the methods with respect to flexibility, ease of use, and precision. Next, in Paper B, we developed a benchmark dataset for stereo-based 3D multi-object tracking of zebrafish in a controlled aquarium environment as illustrated in Figure 1.2. We used the knowledge gained from Paper A to design the hardware setup and as part of our baseline 3D multi-object tracker. Through our work with the zebrafish dataset, we discovered that there was no effective method for estimating the complexity of MOT sequences. This lack of metric obstructs creation of 'fair' data splits and complicates objective comparison between datasets and, thereby, trackers. Therefore, in Paper B we also developed a novel occlusion-based metric to estimate the complexity of 3D fish sequences, as we deemed occlusion to be the largest hindrance for fish tracking. In Paper C, we expanded the work on estimating sequence complexity by proposing a MOT complexity metric, that generalizes to other types of MOT datasets. The metric is based on the level of occlusion, non-linear motion, and visual similarity in the sequences.

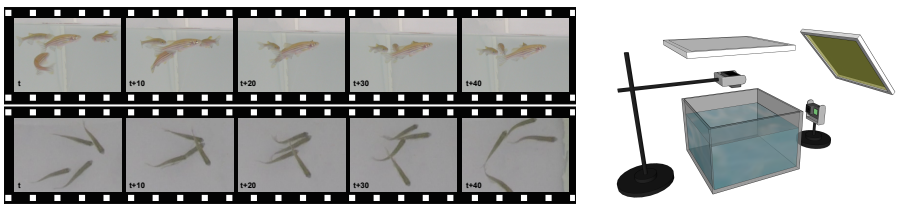


Fig. 1.2: Cut-out examples from the 3D zebrafish tracking dataset proposed in Paper B and the stereo setup used to record the dataset. The figure is from [14], Paper B, © 2020 IEEE.

Chapter 3: Monitoring Marine Organisms in the Wild

In this part of the work, we moved from a controlled laboratory environment and into local coastal waters to focus on detecting and tracking marine organisms in the wild. This led to three publications beginning with Paper D, in which we presented a custom-made camera setup with artificial lights designed for capturing video sequences in turbid water with strong currents. The setup was mounted nine meters below surface and it recorded video sequences for two years laying the foundation for the *Brackish dataset*, a publicly available object detection dataset with annotated bounding boxes. A frame with bounding box annotations from the dataset is presented in Figure 1.3. In Paper E, we expanded upon the Brackish dataset by adding nine new fully annotated sequences focusing on schools of fish. Additionally, the ground truth annotations of all the sequences were improved with IDs to enable multi-object tracking of marine organisms in the wild. We utilized the MOTCOM metric, proposed in Paper C, to design the train and test splits to ensure a fair distribution of sequences with respect to complexity.

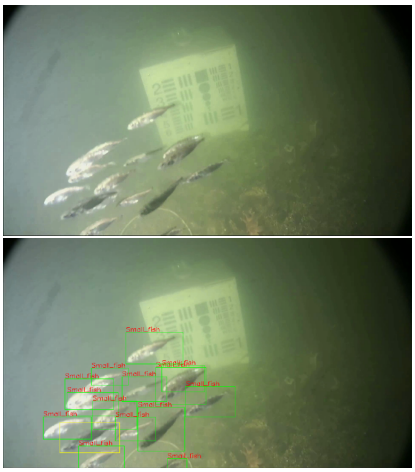


Fig. 1.3: A frame from the Brackish dataset [13] without and with bounding box annotations drawn around the fish. The annotations were updated to include ID's in the BrackishMOT dataset [16]. The figure is adapted from [13], Paper D, © 2019 IEEE.

Data acquisition in underwater environments is notoriously difficult and during development of our underwater camera system we discovered a gap in the existing literature on guidelines for conducting long-term underwater recordings and the precautions to take. Therefore, the experience we gained from designing, mounting, and recording data in a harsh brackish environment serves as the basis for Paper F, which is an essay that describes general factors to consider when collecting data in marine environments.

Chapter 4: Long-term Monitoring of Giant Sunfish

Lastly, we steer into international waters, more specifically Indonesia, where we investigate the problem of monitoring individual giant sunfish across years by re-identifying them based on their unique patterns. Despite its large size, the giant sunfish is elusive and rarely seen. However, they periodically visit coastal areas near Bali in Indonesia and are therefore occasionally pho-

tographed by amateur divers. These images are the basis for on-going research on the state of the giant sunfish and how it is affected by climate change, visits from curious divers, and much more. The images are gathered continuously and stored in a database named 'Match My Mola', which currently holds a few thousand images. The re-identification procedure is currently done manually by trained volunteers through visual inspection as illustrated in Figure 1.4 and this is an extremely time-consuming and tedious task. Therefore, in Paper G we introduce a novel and effective method for recognizing individual giant sunfish based on keypoint matching which requires no training or fine-tuning. In Paper H, we build on top of our previous work, by proposing an enhanced version of the method, and evaluate its performance on two additional datasets of patterned animals. Our results demonstrate the method's generalizability and potential for wider applications based on its ranking capabilities. However, despite that a ranked output is common in academia, it is not necessarily practical for real-life applications where it is not trivial to decide the number of proposed matches to manually verify. To address this issue, we propose an alternative binary output module for the pipeline that states whether two images depict the same individual. This approach is expected to be more suitable for a human-in-the-loop system in which efficient manual verification is critical.

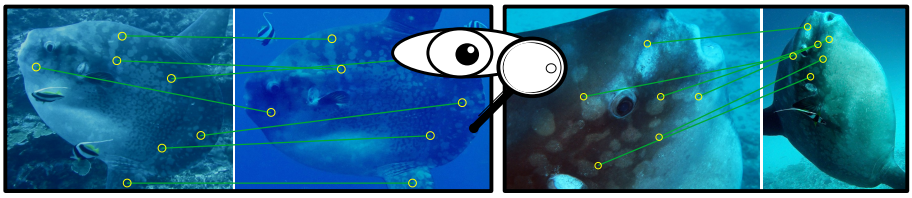


Fig. 1.4: Currently, marine researchers manually re-identify Giant Sunfish by comparing the unique patterns on their bodies. The figure is from [17], Paper H.

Final Remarks

In the following three chapters of Part I we will dive deeper into the work and findings that have been made throughout this project. Every chapter concludes with a summary that emphasizes the main contributions accompanied by a discussion of possible future research directions. This is followed by a Part II, which is a collection of the papers that laid the foundation for the findings.

I hope you will enjoy reading the thesis and maybe learn a bit on the way.

References

[1] "Waymo open dataset: An autonomous driving dataset," 2019, https://www.waymo.com/open.

[2] *The State of World Fisheries and Aquaculture 2022*. FAO, jun 2022.

[3] P. Dendorfer, A. Osep, A. Milan, K. Schindler, D. Cremers, I. Reid, S. Roth, and L. Leal-Taixé, "Motchallenge: A benchmark for single-camera multiple target tracking," *International Journal of Computer Vision (IJCV)*, vol. 129, p. 845–881, 2021.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2009, pp. 248–255.

[5] M. Fabbri, G. Brasó, G. Maugeri, O. Cetintas, R. Gasparini, A. Ošep, S. Calderara, L. Leal-Taixe, and R. Cucchiara, "Motsynth: How can synthetic data help pedestrian detection and tracking?" in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 10 829–10 839.

[6] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2012, pp. 3354–3361.

[7] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "MS-celeb-1m: A dataset and benchmark for large-scale face recognition," in *Computer Vision – ECCV 2016*. Springer, 2016, pp. 87–102.

[8] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The MegaFace benchmark: 1 million faces for recognition at scale," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 25. Curran Associates, Inc., 2012, pp. 1097–1105.

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Computer Vision – ECCV 2014*. Cham: Springer, 2014, pp. 740–755.

[11] N. Madsen, M. Pedersen, K. T. Jensen, P. R. Møller, R. E. Andersen, and T. B. Moeslund, "Fishing with c-tucs (cheap tiny underwater cameras) in a sea of possibilities," vol. 16, no. 2, pp. 19–30, 2021. [Online]. Available: https://www.thejot.net/article-preview/?show_article_preview=1250

[12] M. Nyegaard, "There be giants! the importance of taxonomic clarity of the large ocean sunfishes (genus mola, family molidae) for assessing sunfish vulnerability to anthropogenic pressures." Ph.D. dissertation, Murdoch University, 2018. [Online]. Available: http://researchrepository.murdoch.edu.au/id/eprint/41666

[13] M. Pedersen, J. Bruslund Haurum, R. Gade, and T. B. Moeslund, "Detection of marine animals in a new underwater dataset with varying visibility," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 18–26.

[14] M. Pedersen, J. B. Haurum, S. Hein Bengtson, and T. B. Moeslund, "3d-zef: A 3d zebrafish tracking benchmark dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2423–2433.

[15] M. Pedersen, S. Hein Bengtson, R. Gade, N. Madsen, and T. B. Moeslund, "Camera calibration for underwater 3D reconstruction based on ray tracing using Snell's law," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2018.

[16] M. Pedersen, D. Lehotský, I. Nikolov, and T. B. Moeslund, "BrackishMOT: The brackish multi-object tracking dataset," in *Image Analysis. SCIA 2023.* Springer, 2023, pp. 17–33.

[17] M. Pedersen, M. Nyegaard, and T. B. Moeslund, "Finding nemo's giant cousin: Keypoint matching for robust re-identification of giant sunfish," *Journal of Marine Science and Engineering*, vol. 11, no. 5, 2023.

[18] K. Sun, W. Cui, and C. Chen, "Review of underwater sensing technologies and applications," *Sensors*, vol. 21, no. 23, p. 7849, nov 2021.

[19] United Nations, "Life below water," https://www.un.org/sustainabledevelopment/goal-14-life-below-water/.

[20] ——, "The sustainable development goals report 2022," https://unstats.un.org/sdgs/report/2022/, Department of Economic and Social Affairs, Tech. Rep., 2022.

[21] ——, "World population prospects 2022: Summary of results," https://www.un.org/development/desa/pd/sites/www.un.org.development.desa.pd/files/wpp2022_summary_of_results.pdf, Department of Economic and Social Affairs, Tech. Rep., 2022.

[22] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang, and S. Lyu, "Ua-detrac: A new benchmark and protocol for multi-object detection and tracking," *Computer Vision and Image Understanding*, vol. 193, p. 102907, 2020.

[23] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 40, no. 6, pp. 1452–1464, jun 2018.

[24] P. Zhu, L. Wen, D. Du, X. Bian, Q. Hu, and H. Ling, "Vision meets drones: Past, present and future," 2020.

Chapter 2

Tracking Multiple Fish in a Controlled Environment

Behavioral analysis of animal models, such as the zebrafish, plays an important role in a wide variety of applications ranging from development of drugs to monitoring the state of ecosystems [24, 26, 38, 66]. By analyzing the behavior of animals, insights can be gained into how environmental factors, such as pollution or habitat degradation, impact the overall health of an ecosystem. This information can be used to inform conservation efforts and guide policy decisions aimed at protecting and preserving fragile ecosystems. Additionally, the use of computer vision in behavioral analysis of animal models allows for objective and scalable experiments to meet the rising demand for large-scale studies [64].

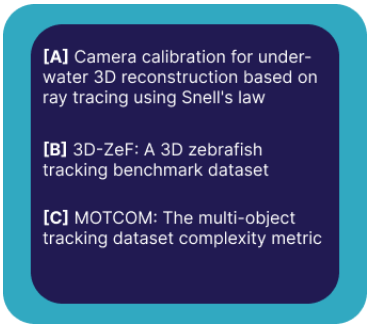


Fig. 2.1: An overview of the papers laying the foundation for the findings in this chapter.

The focus in this chapter is on tracking multiple zebrafish in a controlled environment. An overview of the papers that lays the foundation for the findings in the chapter can be found in Figure 2.1.

1 Laying the Foundation for Precise Underwater 3D Reconstruction of Fish Trajectories

Despite a controlled aquarium environment, it is a complicated task to accurately track multiple similarly looking, erratically moving, and social fish [40].

The first step is to identify an optimal hardware configuration for the aquarium, camera, and light. Solutions for automated behavioral analysis has typically been developed for terrestrial animals like mice or aquatic organisms in shallow water [19, 43, 47, 53, 56] and has focused on tracking objects in 2D using a single camera. However, 2D solutions are not capable of capturing all the relevant motion patterns exhibited by fish as their natural habitats allow for movement in three dimensions [25]. Therefore, an aquarium that allows the fish to swim freely is a prerequisite for conducting meaningful behavioral analysis of fish [36]. Furthermore, in order to estimate the 3D trajectories of the fish, depth information is necessary.

Choosing a Suitable Sensor Setup

There exists a range of sensors and methods for acquiring depth information including structured light, time-of-flight, and stereo vision. Relying on a single depth sensor to capture swimming fish can lead to complete occlusion of objects if the fish swim in front of each other, which is likely due to the social behavior of zebrafish. A way to address this issue, is to use a stereo setup with two cameras and a wide baseline that captures the fish from approximately orthogonal viewpoints, as shown in Figure 2.2. With this setup, if an object is occluded in one view, there is a chance that it is still visible in the other. This reduces the likelihood of complete occlusion, which is commonly considered to be the most critical problem for tracking algorithms to handle. The advantages of multi-camera stereo setups are highlighted in the literature, where it is the most used setup for tracking fish in 3D in controlled environments [10–12, 42, 48, 59, 65, 68]. Therefore, we also choose to use this setup.

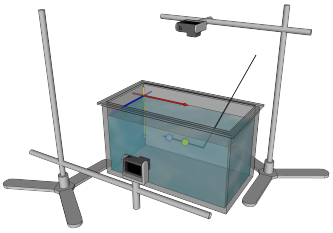


Fig. 2.2: Illustration of our stereo setup with orthogonal views. The rod with the two balls is used for evaluating the calibration procedures and will be explained in more details. The figure is from [46], Paper A, © 2018 IEEE.

Refraction

When the subjects are submerged in water one must consider the effect of refraction, which is when light waves change direction as they move between two media as illustrated in Figure 2.3. Refraction can be described through Snell's law which is given by

$$\frac{\sin\theta_1}{\sin\theta_2} = \frac{v_1}{v_2} = \frac{n_2}{n_1} \tag{2.1}$$

"where θ is the angle between the surface normal and the light ray, v is the velocity of light, and n is the refractive index of the respective medium" - from [46], Paper A,

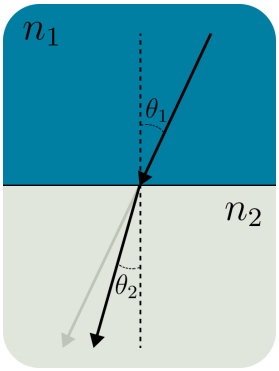


Fig. 2.3: Illustration of the angular change of light due to the movement between two transparent media with different refractive indexes.

This phenomenon complicates the task of obtaining accurate 3D information from depth sensors. Submerging the cameras into the water will reduce some of the distortion caused by refraction. However, it will lead to blind spots where the fish are able to hide, e.g., behind or beside the camera, and the fish may feel threatened or attracted by the intrusive cameras, resulting in undesirable behaviors. Additionally, there will still be a need for calibration in order to acquire accurate depth estimations. Therefore, in most cases it is far from optimal to place the cameras in the water.

The other possibility is to place the cameras outside the aquarium. This induces a stronger distortion caused by refraction as it introduces additional transitions when the light travels from the water through the aquarium wall and into air before reaching the lens of the camera. However, due to the relatively thin walls in smaller aquariums the impact of the glass/acrylic medium is negligible and can be discarded [63]. The refraction between air and water still needs to be taken into account, though, in order to obtain accurate measurements [73]. We choose to place the cameras outside the aquarium to avoid the aforementioned negative consequences of occlusion and intrusion associated with submerging the cameras into the water.

Calibration Procedures

Broadly speaking, there are two categories of calibration procedures used to deal with refraction: relying on the camera model to indirectly absorb the transformations caused by refraction or explicitly taking refraction into account [7, 63].

In the first category the simplest approach is to rely on the single viewpoint (SVP) camera model and calibrate the camera following Zhang's camera calibration method [79], which is included in most image processing toolboxes. Zhang's method is based on capturing images of a planar object (typically a checkerboard) with known dimensions and solving a system of linear equations that relates known 3D point correspondences, such as corners, on the object, with 2D points found in the image. Typically, around 20 images are captured of the checkerboard placed in different positions, orientations, and distances to the camera. By moving the checkerboard in air this procedure

can provide both the intrinsic and extrinsic camera parameters. However, this approach does not in any way account for refraction, therefore, a slightly more advanced procedure is to submerge the checkerboard into the aquarium while capturing the calibration images [8, 48, 57]. The 3D reconstruction accuracy is heavily dependent on the quality of the images of the checkerboard as well as the localization of the 2D-3D point correspondences. Additionally, the cameras must be re-calibrated if they are moved, which is time-consuming and tedious.

In the second category, refraction is taken into account by modelling the physics that causes the light to change direction as it moves between media [63]. In the field of 3D reconstruction of fish trajectories the procedure is typically to compute the intrinsic parameters of each camera through in-air calibration, e.g., following Zhang's method. Subsequently, refraction is considered by estimating the path of the light based on the placement of the cameras with respect to the aquarium and using Snell's law to account for the light's angular change between the media [9, 23, 42, 70, 72]. The extrinsic parameters are estimated in order to establish a correspondence between the cameras and the aquarium in a shared world-space. If the intrinsic parameters are known, the extrinsic parameters can be found by solving the perspective-n-point (PnP) pose problem from a set of fixed calibration points on the aquarium. A flexible setup can be achieved by separating the computation of the intrinsic and extrinsic parameters. This allows for moving the cameras or changing the aquarium without having to re-calibrate the intrinsic parameters, it merely requires a single image of the fixed calibration points on the aquarium (e.g., the corners) in order to re-establish the world-to-image (3D-2D) relationship [46], as explained in Paper A. With the intrinsic and extrinsic parameters in place, the final steps involve projecting the 2D image point into a ray, identify where it intersects with the aquarium wall, and use Snell's law to compute the angular change of the ray between the two media. This procedure is done for both cameras and the 3D position is estimated from triangulation of the two refracted rays. A simplified overview of the steps from camera calibration to triangulation are illustrated in Figure 2.4.

Comparing the Calibration Procedures

There are pros and cons of both types of calibration procedures, but they have not been quantified in the literature. Therefore, in Paper A, we propose an evaluation procedure based on manually moving a rod with two brightly colored balls around an aquarium and subsequently estimating the distance between the two balls, see Figure 2.2 for an illustration of the rod. The difference between the true and estimated distance between the balls is used to estimate the precision of the 3D reconstruction method. We evaluate the three calibration procedures described previously, namely:

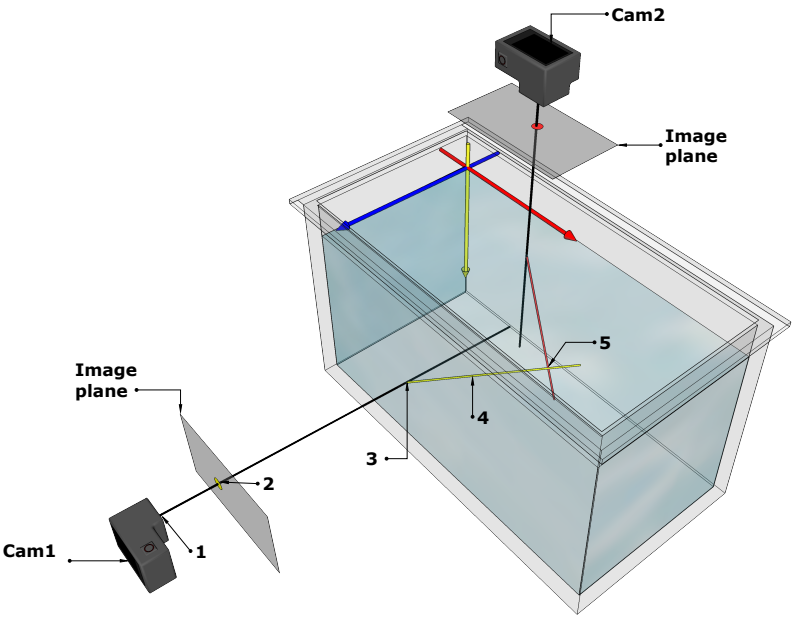


Fig. 2.4: *"Illustration of the five 3D reconstruction steps. 1. Camera calibration 2. Projecting a 2D point into a ray 3. Identifying the plane-ray intersection 4. Calculating the refracted rays 5. Triangulation using rays."* [46]. Figure from [46], Paper A, © 2018 IEEE.

1. *In air*: Estimating the intrinsic parameters using Zhang's method with a checkerboard in air.

2. *Under water*: Estimating the intrinsic parameters using Zhang's method with a checkerboard submerged under water and moved around the aquarium.

3. *Ray tracing*: Estimating the intrinsic parameters following the *in air* calibration and subsequently taking refraction into account explicitly.

Note that for all three methods, the extrinsic parameters are found through solving the PnP problem with four calibration points (the aquarium corners) and the 3D point is estimated by triangulation of the ray-projected 2D image coordinates.

We find that the *in air* procedure introduces large reconstruction errors, which is not surprising as it is not designed to handle refraction. On the other hand, we see that calibrating the intrinsic parameters using the *under water* and *ray tracing* procedures both give good results with low error-margins making both of them suitable for precise 3D reconstruction of fish trajectories in controlled environments. However, the *ray tracing* procedure is found to be far

more flexible due to the decoupling between the intrinsic and extrinsic parameters. When following the *under water* procedure, the intrinsic parameters can to a degree absorb the distortion caused by refraction, but it comes at the expense of a rigid setup. If either the camera or aquarium is moved the entire procedure must be repeated. Additionally, the user must ensure that the checkerboard is moved around the entire aquarium during calibration as it otherwise can lead to regions with lower reconstruction precision. We grade the procedures by three loosely defined attributes: precision, flexibility, and precision and the findings are summarized in table Table 2.1.

	In air	**Under water**	**Ray tracing**
Precision	✓	✓✓✓	✓✓✓
Flexibility	✓	✓	✓✓✓
Ease of use	✓✓✓	✓	✓✓

Table 2.1: *"Simplified recap and comparison of the properties of the tested 3D reconstruction approaches."* [46]. Table is from [46], Paper A, © 2018 IEEE.

2 Tracking Multiple Zebrafish in 3D: Developing a Benchmark Dataset and Tracker

The majority of multi-object trackers has been developed for pedestrians and traffic scenes [4, 37, 75, 77, 78, 80], likely due to its proximity and relevance to our daily lives, and this has naturally led to the development of multiple large public benchmark datasets [13, 14, 20, 76]. Despite that pedestrians, cars, and bicycles move in three dimensions, the tracking task is most often simplified to an estimation of 2D trajectories due to the movement taking place on an approximately planar surface. This is obviously not the case for fish, nonetheless, a majority of trackers used for behavioral analysis of fish in controlled environments has been developed for 2D tracking [41, 47, 49, 50, 55, 56, 58]. This has led to a frequent use of aquariums with only a few centimeters of water, which forces the fish to follow an approximate 2D plane. However, by limiting the space, a range of behavioral traits disappear and the outcome becomes only a partial behavioral analysis [10, 25, 36]. Therefore, it is paramount to observe fish in three dimensions, but this poses additional requirements to the tracker.

Most 3D fish tracking algorithms have been developed specifically for zebrafish [3, 10, 48, 59, 70, 72] due to zebrafish being a commonly used animal model [17, 25, 26, 31]. For this reason and for comparability, we also chose zebrafish for our work in Paper B. The literature on this subject generally present trackers with, at first glance, surprisingly strong performance in 3D tracking

of 10-20 individual zebrafish [33, 48, 51, 67, 68]. However, their evaluation procedures contain biases and shortcomings and most likely do not reflect real-world conditions. For example, Wang et al. [67, 68] do not present or describe their training data. Additionally, they claim to make their dataset available, however, despite multiple attempts we were not able to obtain it. Another example is Qian et al. [33, 48, 51] who evaluate their algorithms on two test sequences, but, like Wang et al., fail to properly explain their training data; their description even hints that they are using a subset of frames from the test sequences as training, but the details are unclear. Furthermore, they present a demo video with 10 fish which contains only four occlusion events during 15 seconds. For comparison, in the dataset we propose, our test sequence with 10 fish contains 66 occlusion events during 15 seconds. Together, these limitations call into question both the difficulty of the scenarios on which the algorithms were evaluated and, more critically, the credibility of the reported results. When evaluating the performance of trackers (or any other type of algorithm for that matter) it is critical to ensure that they are tested on a diverse range of conditions in order to provide a trustworthy estimate of their effectiveness in real-world scenarios. Therefore, we chose to develop our own 3D zebrafish tracking dataset and make it publicly available for others to evaluate their trackers on a common benchmark.

The 3D-ZeF Benchmark Dataset

During development of the 3D zebrafish tracking dataset (3D-ZeF) we aimed at being apparent about every aspect regarding setup, recording procedures, and data splits, as this was heavily lacking in the field. Additionally, the setup should be replicable, therefore, we deliberately chose to use off-the-shelf equipment with a simple square glass aquarium, two IKEA lights, and two GoPro cameras positioned as illustrated in Figure 2.5. A detailed description of the recording setup can be found in Paper B.

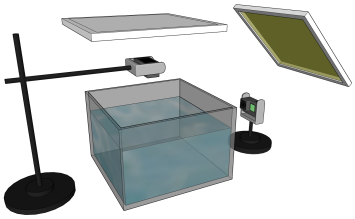


Fig. 2.5: Illustration of the stereo setup with a square aquarium, orthogonal views, two lights. The figure is adapted from [45], Paper B, © 2020 IEEE.

Beside deciding on the setup, we had to determine which and how many fish should be present in the sequences. Dependent on where and how zebrafish are raised, how old they are, and their general state, they may look or behave slightly different. Therefore, it was critical for us that the fish in each of the three data splits (train, validation, and test) were unique to avoid data leakage and minimize the chance of overfitting. Additionally, we chose to use fish from an entirely different, and younger, population for the test split compared to the fish from the train and validation

	Trn2	Trn5	Val2	Val5	Tst1	Tst2	Tst5	Tst10	Total
Length	120 s	15 s	30 s	15 s	15 s	15 s	15 s	15 s	240 s
Frames	14,400	1,800	3,600	1,800	1,800	1,800	1,800	1,800	28,800
BBs	28,800	9,000	7,200	9,000	1,800	3,600	9,000	18,000	86,400
Points	28,800	9,000	7,200	9,000	1,800	3,600	9,000	18,000	86,400
OC	1.82 / 1.42	3.60 / 2.93	0.93 / 0.47	2.67 / 3.80	0.00 / 0.00	0.67 / 0.67	3.07 / 2.93	4.40 / 6.53	
OL	0.41 / 0.51	0.56 / 0.64	0.22 / 0.63	0.25 / 0.66	0.00 / 0.00	0.10 / 0.38	0.25 / 0.36	0.28 / 0.35	
TBO	0.69 / 0.89	1.00 / 1.21	1.79 / 3.20	1.64 / 0.73	15.00 / 15.00	2.41 / 2.18	1.38 / 1.28	1.86 / 1.40	
IBO	0.29 / 0.26	0.28 / 0.28	0.24 / 0.35	0.22 / 0.34	0.00 / 0.00	0.19 / 0.19	0.25 / 0.23	0.26 / 0.24	
Ψ	**0.26**	**0.50**	**0.03**	**0.63**	**0.00**	**0.01**	**0.16**	**0.28**	

Table 2.2: *"Overview of the proposed dataset. OC, OL, TBO, and IBO are listed for the top- and front-view, respectively, and the number of fish is denoted in the sequence name. OC: average amount of occlusions per second, OL: average occlusion length in seconds, TBO: average amount of seconds between occlusions, IBO: intersection between occlusions, Ψ: complexity measure based on OC, OL, TBO and IBO (see Equation (2.2)).*" [45], table from [45], © 2020 IEEE.

splits, which were from the same group. This was a significantly different approach compared to the procedures followed in the literature.

In total we recorded eight sequences which were split into two train, two validation, and four test sequences with 1-10 individuals, the details can be found in Table 2.2. Each sequence consists of two synchronized videos: one captured from the camera placed in front of the aquarium (front) and another captured from above the aquarium (top). All videos were fully and manually annotated with bounding boxes (BBs) and points. The points were used as the groundtruth for the 3D tracks to be reconstructed. The only rigid part of a zebrafish is its head, therefore, the point to track and reconstruct was positioned on top of the head between the eyes (when seen from the top view) or in one of the eyes (when seen from the front view) as illustrated in Figure 2.6.

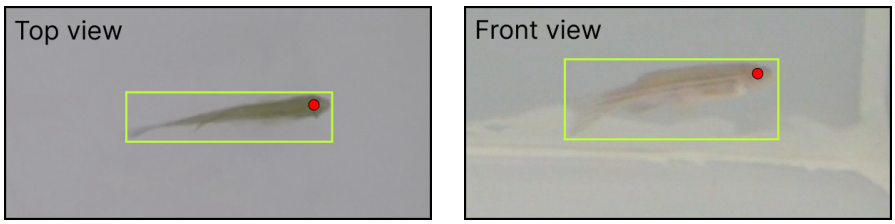


Fig. 2.6: Examples of zebrafish seen from the top and front view with ground truth bounding boxes and points drawn on the images. Note that the images are cropped in proximity to the objects and are from larger frames containing multiple zebrafish.

During the annotation process we realized that the complexity of a sequence did not seem to be dependent on the number of individuals, but rather on their behavior. In some cases when the fish occluded each other we had to switch between the two views repeatedly to confirm our annotations, while the annotation process was unambiguous and fast when the fish were spatially distant. This led us to develop a score for estimating the complexity of the sequences based on the level of occlusion. The complexity scores are presented

as part of Table 2.2 for both views of each sequence. The overall complexity score, Ψ, is computed from the four sub-metrics presented below along with a brief justification for their existence:

- **OC** (Occlusion count): More occlusions makes a harder problem.
- **OL** (Occlusion length): Longer occlusions makes a harder problem.
- **TBO** (Time between occlusions): Less time between occlusions makes a harder problem.
- **IBO** (Intersection between occlusions): A higher degree of occlusion makes a harder problem.

The final score was then computed as:

$$\Psi = \frac{1}{n} \sum_{v}^{\{\mathrm{T,F}\}} \frac{\mathrm{OC}_v \, \mathrm{OL}_v \, \mathrm{IBO}_v}{\mathrm{TBO}_v}, \tag{2.2}$$

Interestingly, the Trn2 sequence (with two fish) has an occlusion-complexity score of 0.26 resembling the score of sequence Tst10 (with 10 fish), which has a score of 0.28. Additionally, Val5 and Tst5 have scores of 0.63 and 0.16, respectively, despite both containing five fish. As we anticipated, this indicates that the number of objects alone is not a well-suited metric for estimating the complexity of tracking sequences.

In addition to the dataset and the occlusion-complexity metric we also proposed a baseline 3D multi-object tracker, which is described in the following section.

A Baseline Tracker for 3D Zebrafish Tracking

When using a stereo camera setup to track objects in 3D, traditionally, there have been two main approaches: reconstruction-tracking and tracking-reconstruction [71]. In the reconstruction-tracking approach, 3D positions of objects are reconstructed based on detections in both camera views for each frame, which is followed by a temporal association of the 3D points to create tracks. This approach benefits from the use of 3D information during the association step, which is generally more accurate. However, weak or missing 2D detections can lead to a complex 3D reconstruction phase.

The tracking-reconstruction approach, on the other hand, starts by solving a traditional 2D MOT problem independently for each camera view. Tracks are then associated between the views, and finally, 3D positions are reconstructed based on the associated tracks. This approach has the advantage of using tracks rather than individual detections for the between-views association step. However, flaws in the 2D tracks, e.g., introduced by ID swaps, can

be problematic to handle during the reconstruction phase. Despite this, we decided to adopt the tracking-reconstruction approach as the basis of our 3D tracker which is presented in Paper B. One of the reasons behind this choice is that the field of MOT is mainly driven by improvements in 2D tracking [4, 77, 78, 80], and by designing our 3D tracker in a modular manner, it is trivial to replace the 2D tracking module as needed [6].

The pipeline of the proposed tracker consists of four modules: 2D object detection, 2D tracklet[1] construction, 2D tracklet association between views, and 3D tracklet association. The groundtruth data used for evaluating the performance of our tracker consists of tracks that follows the head of the fish. Therefore, the objective of the tracker is to locate, track, and reconstruct the 3D positions of the zebrafish head.

- **2D object detection:** We implement two different approaches: a Faster R-CNN [52] head detector and a naive BLOB detector with a skeletonization step. The head detector has the advantage that as long as the head of the fish is visible, occlusion is potentially not a problem. The naive detector is based on background subtraction, which is a robust segmentation algorithm to use in controlled environments that requires minimal fitting, however, it does not handle occlusions well.

- **2D tracklet construction:** The Euclidean distances between the 2D detections are used to construct a cost matrix, which is solved as a global optimization problem through the use of the Hungarian algorithm [28]. We deliberately choose a conservative approach to obtain more robust tracklets at the expense of their length.

- **2D tracklet association between views:** This module utilizes the *ray tracing* calibration and triangulation procedure presented in Paper A for reconstructing 3D positions. The problem of associating the 2D tracklets is formulated as a directed acyclic graph. The graph is created by analyzing each tracklet in the top-view and associating it with front-view tracklets that intersect temporally. Every node describes a 3D tracklet that is composed by one 2D tracklet from both views. The weight of a node is computed as a temporal median of the reprojection error of the per-frame reconstructions between the 2D tracklets. Every node is connected to all other nodes sharing *one* of the 2D tracklets under the condition that connected tracklets from the same view cannot temporally overlap as exemplified in Figure 2.7. The weight of the edges are based on the number of frames between the last frame of the first node and the first frame of the second node, the speed of the fish, and the weights of the two nodes. The graph is disconnected, meaning that a node can be

[1]We use the word 'tracklet' for a trajectory that does not necessarily contain the full spatio-temporal information of an object's motion. In other words; a partial track.

included without being connected to other nodes. Lastly, the 3D tracklets are found by greedily choosing the path through the graph that gives the highest score when summing the weights of the nodes and edges. Every time a tracklet is constructed the associated nodes and edges are removed, and this is continued in a recursive manner until the graph is empty as illustrated in Figure 2.8.

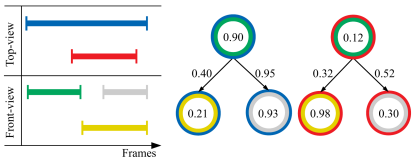


Fig. 2.7: *"The colored lines represent 2D tracklets in each view, the node pairs are represented by the double-colored circles, and the edges of the DAG are shown by the arrows. The numbers represent example node and edge weights."* [45], the figure is from [45], Paper B, © 2020 IEEE.

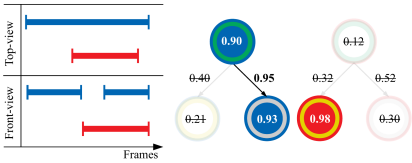


Fig. 2.8: *"Graph evaluation based on the example from Figure 4. The colored lines represent 2D tracklet pairs based on the chosen nodes in the graph; the transparent nodes are discarded."* [45], Paper B, © 2020 IEEE.

- **3D tracklet association:** The last step is to associate the 3D tracklets into the final tracks. The number of fish in the aquarium, N, is known and constant throughout every sequence. We utilize this to initiate N **main** tracks, which are found by looking for sets of N temporally overlapping tracklets and choosing the set containing the most robust candidates. The remaining tracklets are then associated with the **main** tracks in a greedy manner based on the spatio-temporal distance between them. This is continued until there are no more tracklets available and the outcome is the N final 3D tracks.

We evaluated our tracker with respect to a range of MOT metrics including MOTA [5], which was the dominant multi-object tracking performance metric at the time. We considered a detection to be a true positive if it was within 0.5 cm from the ground truth annotation. Beside evaluating the pipeline with the Faster R-CNN and naive detectors, we also presented results from an 'oracle tracker'. The oracle tracker simply tracks everything correct as long as the object is not part of an occlusion in any of the two views. This gives us a strong basis of comparison with three trackers with very different foundations. The results from the three trackers are presented in Table 2.3.

We see that the naive detector is more stable compared to the Faster R-CNN based detector. The Faster R-CNN model may have overfitted to the zebrafish from the train split and has not managed to generalize to other zebrafish. Whether this is the case may be interpreted from the plot presented in Figure 2.9, which shows the performance of each of the trackers with respect to

	Method	MOTA ↑	MT ↑	ML ↓	ID Sw. ↓	Frag. ↓	MTBF_m ↑
Tst1	Naive	77.6%	1	0	0	28	12.5
	FRCNN-H	30.2%	0	0	0	15	8.212
	Oracle	100.0%	1	0	0	0	900
Tst2	Naive	77.6%	1	0	0	44	15.856
	FRCNN-H	5.7%	0	2	2	17	2.641
	Oracle	81.6%	2	0	0	25	27.396
Tst5	Naive	39.7%	0	0	7	185	6.249
	FRCNN-H	40.2%	0	0	7	115	7.577
	Oracle	67.8%	1	0	0	50	28.112
Tst10	Naive	48.3%	0	0	11	268	9.075
	FRCNN-H	25.2%	0	3	32	225	4.904
	Oracle	66.6%	1	10	0	119	23.105

Table 2.3: *"Evaluation of 3D tracks on test split. The arrows indicate whether higher or lower values are better. MOTA: Multiple Object Tracking Accuracy, MT: Mostly tracked, ML: Mostly lost, ID Sw.: Number of identity swaps, Frag.: Number of fragments, MTBF_m: Monotonic MTBF."* [45], table is from [45], Paper B, © 2020 IEEE.

the occlusion-complexity score for all the sequences from the train, validation, and test splits.

Interestingly, we see that the Faster R-CNN based tracker performs well on both the training and validation sequences, but poorly on the test sequences. This is in accordance with the assumption that the detector has been overfitted as the train and validation splits contain fish from the same population, whereas the test sequences contain younger fish from an entirely different population. The fact that the Faster R-CNN based tracker significantly outperforms the oracle tracker on three out of four of the training and validation sequences, indicates that the tracking and reconstruction modules work well with a strong detector. Lastly, we see that the performance of the naive tracker correlates negatively with the occlusion-complexity metric, which is expected qua the design of the detector.

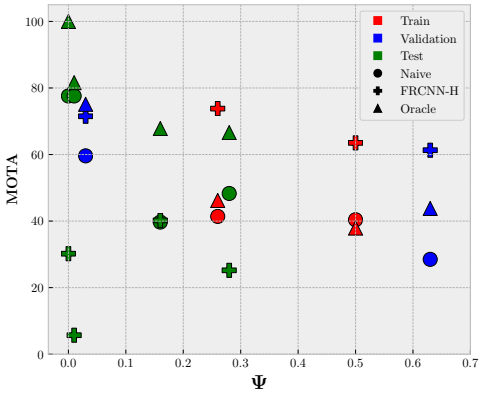


Fig. 2.9: *"MOTA compared to the dataset complexity, Ψ, for all sequences in the dataset."* [45], figure is from [45], Paper B, © 2020 IEEE.

3 Estimating the Complexity of Multi-Object Tracking Sequences

The purpose of the complexity metric presented in Paper B was to allow for a more fair comparison between zebrafish trackers and datasets. However, a metric that relies solely on occlusion fails to capture the various challenging aspects of more general MOT problems. Furthermore, using the metric only for comparison purposes would not fully exploit its potential. A comprehensive complexity metric could provide more transparency on tracker performance in general and be utilized to tailor dataset splits based on informative criteria. Therefore, in Paper C, we steered into uncharted waters on a pursuit for developing a comprehensive MOT dataset complexity metric.

The literature in the field of MOT dataset complexity estimation is extremely sparse. Traditionally, the complexity of MOT sequences has been evaluated in terms of the number of objects, and this has been reflected in the construction of datasets, which has generally aimed to collect sequences with progressively more objects [15, 18, 29, 39, 61]. However, the idea of designing dataset splits based on their complexity has likely been around for decades, but it has been based on subjective estimates by the developers and not on common terms. This is exemplified in PETS2009 [18] where the authors provide a subjective diffculty score for the sequences, without going further into details. For the popular MOTChallenge datasets [14], the developers provide sequence information like frame rate, resolution, and length. Additionally, they present two metrics based on the number of objects, namely, the total *number of tracks* and the average number of objects per frame denoted *density*. Leal et al. [30] touches upon some of the aspects that make MOT sequences difficult to handle and which has laid the foundation for the creation of the MOT15 and MOT16 benchmarks. Traits like camera motion, low illumination, and inclusion of night-scenes are used to describe the difficulties of MOT15. For the next iteration of the benchmark, MOT16, the authors specifically mention a focus on increasing the difficulty of the sequences through a higher density of objects.

In Paper C we have dared to challenge this widely held belief that a higher number of objects in a multi-object tracking sequence automatically translates to a greater level of difficulty. We anticipate, based on our experiences from Paper B, that factors derived from the activities in the scene generally have a higher impact on the MOT complexity compared to the number of objects. Therefore, from the MOT literature we identified three major challenges that multi-object trackers aim to address, namely: occlusion, erratic motion, and visual similarity [1, 2, 4, 35, 45]. The three elements are the core of the MOT complexity metric proposed in Paper C. The following sections provide a brief description of each element and how to quantify them in order to estimate the complexity of a sequence.

Hidden in Plain Sight:
- Quantifying the Complexity of Occlusion in MOT Sequences

Occlusion is a term used in MOT to describe when an object is hidden from view, either fully or partially. The literature generally distinguish between three types of occlusion: self-occlusion, scene-occlusion, and inter-object-occlusion [1].

Self-occlusion, such as when a person covers their face with their hands, can be a problem in surveillance tasks, but it is not generally considered a significant problem in MOT. The importance of different parts of an object can vary greatly depending on the type of object and task, which makes self-occlusion difficult to quantify. For example, highlighting one limb may lead to occlusion of another, e.g., if a person raises their arm to expose the torso; depending on the view, the shoulder or face may become hidden. Therefore, we have chosen not to include it in our complexity metric.

In a broader picture scene-occlusion and inter-object-occlusion refer to the same problem: an object being hidden from view due to external forces. However, there is a difference between the two, which for some tasks can be more or less significant, especially for trackers that do not utilize visual cues. Inter-object-occlusion entails close and possibly passing tracks, which typically poses a more difficult problem compared to scene-occlusion where an object is hidden due to moving behind a stationary object like a tree or a sign. We account for both types of occlusion, but do not discriminate between them for simplicity.

The MOTChallenge annotation standard is likely the most widely used and it entails annotating the entire object per frame; even during occlusions. It also includes a *visibility* score which is automatically computed from the ground truth annotations as the area of intersection over the area of the object [14]. During inter-object occlusions the ordering of the objects (who is occluding who) is determined from a pseudo-depth estimation based on the y-values of the bounding boxes, which applies due to an assumption of the objects moving on a plane. We compute our occlusion metric, OCOM, as the complimentary to *visibility*. OCOM is in the interval $[0, 1]$ where a higher number indicates a more complex problem.

Note that we did not use the visibility score as the foundation for the occlusion-complexity metric in 3D-ZeF, Paper B, as we followed another annotation procedure which did not allow us to directly compute the visibility score. The flexible body of the zebrafish makes it a non-trivial task to annotate the entire fish when it is occluded, therefore, only the visible parts of the fish were bounding box annotated in 3D-ZeF. Furthermore, fish moves in three dimensions, therefore, the pseudo-depth estimation based on the y-values of the bounding boxes does not uphold.

Twists and Turns:
- Quantifying the Complexity of Motion in MOT Sequences

In multi-object tracking, motion is a broad term that generally covers an object's change in state between consecutive frames. For most MOT tasks the perceived motion is typically limited to the motion captured on the image plane, rather than the actual 3D motion of the object. In addition to the object's own motion, this can also involve camera motion, passive motion, or a combination of these factors. As more variables come into play, predicting the next state of the object becomes more challenging. Except from the special case of having no motion, linear motion is the simplest to predict. Therefore, we expect objects to behave linearly between consecutive frames and the level of deviation is an indication of erratic behavior and a more complex motion to predict.

Beside the motion itself, the camera placement and perspective also affect how motion is perceived. From bird's-eye view (e.g., traffic-surveillance from a drone) objects are more likely to (approximately) maintain the same distance to the camera, therefore, if an object is moving with a given velocity, this will not significantly change during the sequence. From a first-person perspective, this is not true unless the object is moving perpendicular to the view. Otherwise, if an object is moving towards or away from the camera, the motion is generally perceived to be slower and it either increases or decreases, respectively. Furthermore, as the distance between the object and camera changes, the size of the object is also perceived to change and this further increases the difficulty of predicting where the object is heading [4].

An illustrative overview of the aforementioned problems are presented in Figure 2.10. We take each of the elements into account in the proposed motion complexity metric, MCOM, by formulating it as the size-compensated non-linear displacement between consecutive frames weighted by a log-sigmoid function to get an output in the interval [0,1] where a higher number indicates a more complex problem.

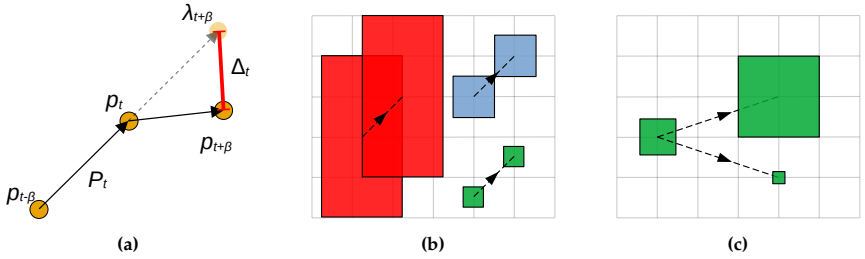


Fig. 2.10: *"a) Illustrative example of how the positional error Δ_t is calculated as the distance between the true position $p_{t+\beta}$ and estimated position $\lambda_{t+\beta}$. b) The three objects have traveled an equal distance. Relative to their size, the two smaller objects are displaced by a larger amount and the bounding box overlap disappears. c) If the size of an object increases between two time steps the displacement is relatively less important, compared to when the size of the object decreases."* [44], the figures are from [44], Paper C.

Spot the Difference:
- Quantifying the Complexity of Visual Similarity in MOT Sequences

Visual cues have been used as an affinity measure for associating detections in tracking algorithms for decades [21, 27, 62]. However, with the advancement of deep learning-based detectors and descriptors, more trackers are placing greater emphasis on visual cues [32, 69, 74, 78]. Generally, tracking becomes more challenging when objects share similar visual characteristics, particularly in scenarios involving occlusions, crowded environments, or fast and erratic motion of the objects among each other. If the objects are spatially distant, however, they are less likely to be confused with one another despite sharing visual characteristics. In other words, the significance of having strong visual cues decreases with the increase in the distance between the objects. Therefore, to estimate the complexity of visual association in MOT sequences we propose a novel method that incorporates both spatial and visual information.

The method involves a preprocessing, feature extraction, and evaluation step which in the end leads to a single visual complexity score, VCOM. Like OCOM and MCOM, VCOM is in the interval [0,1] where a higher number indicates a more complex problem. The preprocessing step consists of heavily blurring the entire image, except for the bounding box region of the target object, which is left unprocessed as illustrated in Figure 2.11. Next, features of the entire image are extracted using an ImageNet [16] pretrained ResNet-18 [22] model. This way, the spatial information is directly embedded into the visual feature vector. Lastly, we measure the distance between the object's features in the current frame and the features of each of the objects in the following frame. The nearest neighbour is expected to have the same ID as the target, however, this is not necessarily the case if the objects are spatially and visually close. A higher number of 'wrong' feature vectors in the proximity of the target feature vector is quantified as an increasingly complex visual problem.

Fig. 2.11: By blurring the entire image, except the object's bounding box, and subsequently computing the object's feature vector from the entire image, we are able to directly embed spatial information into the visual feature representation. The figure is from [44], Paper C.

MOTCOM: An Accurate MOT Dataset Complexity Estimator

Each of the aforementioned sub-metrics describe an element that can make MOT tasks difficult to solve. However, heavy occlusion is not necessarily difficult to handle if the objects have strong visual characteristics that allow for robust re-identification or if the objects move in a predictable manner. Likewise, erratic motion is easily handled if there is no occlusion and the objects are visually distinct. In other words, the complexity of MOT sequences is dependent on multiple factors. Therefore, we propose an overall complexity metric named MOTCOM, which is an average of the three sub-metrics.

Estimating the complexity of a MOT sequence is one thing, verifying whether the estimation is valid is another. There exists no 'groundtruth' for the complexity of a MOT sequence. Currently, the only objective complexity metrics are based on the number of objects (the total number of *tracks* and the average number of objects per frame, called *density*) but as we show in Paper C the number of objects (alone) does not induce a significant problem for multi-object trackers. Therefore, we approximate the groundtruth complexity from the performance of state-of-the-art trackers under the assumption that the complexity has a strong negative correlation with tracker performance.

There are several linked problems that complicates this approach: 1) the MOTCOM metrics can only be computed for sequences where the groundtruth annotations are available, 2) trackers are evaluated on the test split, 3) the major benchmark datasets do not publish groundtruth annotations for the sequences of the test split, and 4) it is not feasible to get an accurate performance estimation by re-implementing trackers as they would have to be both trained and evaluated on the train sequences.

However, with permission from the MOTChallenge team, we were allowed to compute MOTCOM for the test sequences of the MOT17 and MOT20 datasets. This provided us access to both MOTCOM and the performance of hundreds of state-of-the-art pedestrian trackers on two of the most widely used MOT datasets. In Figure 2.12 we present a Spearman's correlation matrix were we compare MOTCOM to *tracks* and *density* with respect to HOTA [34], MOTA [60], and IDF1 [54]. We see MOTCOM has a strong negative correlation with tracker performance while this is not true for *density* and *tracks*. This consolidates that MOTCOM is a better alternative for estimating the complexity of MOT sequences compared to the number of objects.

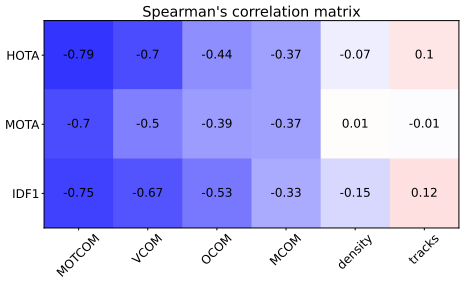


Fig. 2.12: *"Spearman's correlation matrix based on the performance of the top-30 trackers on MOT17 and MOT20."* [44], figure from [44], Paper C.

4 Summary and Scientific Contributions

The focus of this chapter was multi-object tracking of fish in a controlled environment. Fish moves freely in three dimensions, therefore, depth information is essential in order to obtain accurate trajectories. In Paper A we implemented, described, and objectively compared three methods for underwater camera calibration and 3D reconstructing, namely: the *in-air*, *under water*, and *ray tracing* procedures. We found the *ray tracing* procedure to be advantageous to the others, especially with respect to flexibility and precision. Therefore, we used the *ray tracing* calibration and reconstruction procedure as part of the 3D multi-object tracker presented in Paper B.

Beside proposing a 3D tracker, we also developed the first publicly available and fully bounding box and point annotated 3D MOT dataset with fish named 3D-ZeF in Paper B. Additionally, we found a lack of transparency in the zebrafish tracking literature, especially with respect to data-handling, and a common argument referring to the number of objects as being decisive for the difficulty of tracking sequences. However, we argue that the number of objects alone does not pose a problem for most trackers and instead, the level of occlusion should be considered the most critical problem to handle in zebrafish tracking. Following this argument, we proposed a novel occlusion-based metric to describe the complexity of MOT sequences.

We expanded upon the idea of estimating the complexity of MOT sequences in Paper C by reformulating and generalising the occlusion complexity metric and proposing two additional metrics to quantify erratic motion and visual similarity. The three metrics were combined into an overall MOT dataset complexity metric named MOTCOM. We showed that MOTCOM is significantly better at estimating the complexity of the sequences of the popular MOT17 and MOT20 datasets compared to two commonly used metrics based on the number of objects.

The main scientific contributions in this chapter can be summarized as:

- In Paper A we presented the first objective comparison between three common procedures of camera calibration and 3D reconstruction with respect to flexibility, ease of use, and precision.
- In Paper B we proposed the first underwater 3D MOT benchmark dataset together with a novel 3D zebrafish tracker and an occlusion complexity estimation metric.
- In Paper C we expanded upon the MOT complexity estimation metric of Paper B by improving the occlusion metric and including two additional metrics for describing the visual similarity and erratic motion of MOT sequences. We combined the sub-metrics into the first ever objective and accurate MOT dataset complexity metric named MOTCOM.

Furthermore, all code and data from the three papers are publicly available, except for the groundtruth annotations of the MOT17, MOT20, and 3D-ZeF test splits.

Future Work

Paper A describes and evaluates three methods for reconstructing 3D points of objects submerged into water from a stereo setup. We only looked at flat refractive surfaces, but it is not uncommon that analysis of fish are conducted in, e.g., circular shaped aquariums. A next step could, therefore, be to describe and evaluate relevant calibration methods to handle spherical refractive surfaces.

In Paper B we presented a stereo based multi-object tracker and there are several steps that may improve the method. A straightforward enhancement of the proposed pipeline would be to implement and compare a range of state-of-the-art object detectors as we found this to be a critical part of the pipeline where it is possible to pick low-hanging fruits. Additionally, as the reprojection error in the reconstruction step is an indicator for the precision of the 2D detections, it may be used actively to optimize detections and even estimate the position of occluded and undetected objects. In cases where there are no (or weak) detections in one view, a possible method could be to project a set of rays into an anticipated location of the fish and identify the ray that yields the smallest reprojection error when associated with the detection in the opposite view.

The MOTCOM metric, which describes the complexity level of MOT sequences, was proposed in Paper C. It was evaluated on a popular, but relatively limited field of MOT, namely pedestrian tracking. It would be relevant to increase the foundation of the scores by including a more thorough evaluation that spans different fields of MOT in order to cement the score's generalizability. Furthermore, it would be highly interesting to conduct experiments where a MOT dataset is split based on the MOTCOM metrics and evaluate how it affects the performance of trackers. For example, 1) create a simple train split and a complex test split, or vice versa, 2) create a train split with heavy occlusion and a test split with minimal occlusion, or vice versa, 3) and so on. This may reveal to what degree dataset composition affects the performance of models developed to solve a given problem.

References

[1] A. Andriyenko, S. Roth, and K. Schindler, "An analytical formulation of global occlusion reasoning for multi-target tracking," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*. IEEE, 2011, pp. 1839–1846.

[2] A. Andriyenko and K. Schindler, "Multi-target tracking by continuous energy minimization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1265–1272.

[3] G. Audira, B. Sampurna, S. Juniardi, S.-T. Liang, Y.-H. Lai, and C.-D. Hsiao, "A simple setup to perform 3D locomotion tracking in zebrafish by using a single camera," *Inventions*, vol. 3, no. 1, p. 11, Feb. 2018.

[4] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2019, pp. 941–951.

[5] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, pp. 1–10, May 2008.

[6] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3464–3468.

[7] G. Bianco, A. Gallo, F. Bruno, and M. Muzzupappa, "A comparative analysis between active and passive techniques for underwater 3d reconstruction of close-range objects," *Sensors*, vol. 13, no. 8, pp. 11 007–11 031, 2013.

[8] G. Bianco, M. T. Ekvall, J. Bäckman, and L.-A. Hansson, "Plankton 3d tracking: the importance of camera calibration in stereo computer vision systems," *Limnology and Oceanography: Methods*, vol. 11, no. 5, pp. 278–286, 2013.

[9] P. D. V. Buschinelli, G. Matos, T. Pinto, and A. Albertazzi, "Underwater 3d shape measurement using inverse triangulation through two flat refractive surfaces," in *OCEANS 2016 MTS/IEEE Monterey*, 2016, pp. 1–7.

[10] J. Cachat, A. Stewart, E. Utterback, P. Hart, S. Gaikwad, K. Wong, E. Kyzar, N. Wu, and A. V. Kalueff, "Three-dimensional neurophenotyping of adult zebrafish behavior," *PLOS ONE*, vol. 6, no. 3, pp. 1–14, 2011.

[11] X. E. Cheng, S. H. Wang, and Y. Q. Chen, "3D tracking targets via kinematic model weighted particle filter," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2016, pp. 1–6.

[12] X. E. Cheng, S. S. Du, H. Y. Li, J. F. Hu, and M. L. Chen, "Obtaining three-dimensional trajectory of multiple fish in water tank via video tracking," *Multimedia Tools and Applications*, vol. 77, no. 18, pp. 24 499–24 519, Feb. 2018.

[13] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016.

[14] P. Dendorfer, A. Osep, A. Milan, K. Schindler, D. Cremers, I. Reid, S. Roth, and L. Leal-Taixé, "Motchallenge: A benchmark for single-camera multiple target tracking," *International Journal of Computer Vision (IJCV)*, vol. 129, p. 845–881, 2021.

[15] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "Mot20: A benchmark for multi object tracking in crowded scenes," 2020.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2009, pp. 248–255.

[17] J. S. Eisen, "Zebrafish make a big splash," *Cell*, vol. 87, no. 6, pp. 969–977, Dec. 1996.

[18] J. Ferryman and A. Shahrokni, "Pets2009: Dataset and challenge," in *IEEE international workshop on performance evaluation of tracking and surveillance*. IEEE, 2009, pp. 1–6.

[19] J. E. Franco-Restrepo, D. A. Forero, and R. A. Vargas, "A review of freely available, open-source software for the automated analysis of the behavior of adult zebrafish," *Zebrafish*, vol. 16, no. 3, Jun. 2019.

[20] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2012, pp. 3354–3361.

[21] B. Han and L. Davis, "Object tracking by adaptive feature extraction," in *Proceedings of the International Conference on Image Processing (ICIP)*, vol. 3, 2004, pp. 1501–1504 Vol. 3.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016, pp. 770–778.

[23] S. Henrion, C. W. Spoor, R. P. M. Pieters, U. K. Müller, and J. L. van Leeuwen, "Refraction corrected calibration for aquatic locomotion research: application of snell's law improves spatial accuracy," *Bioinspiration and Biomimetics*, vol. 10, no. 4, 2015.

[24] J. Ji, J. Huang, N. Cao, X. Hao, Y. Wu, Y. Ma, D. An, S. Pang, and X. Li, "Multiview behavior and neurotransmitter analysis of zebrafish dyskinesia induced by 6ppd and its metabolites," *Science of The Total Environment*, vol. 838, p. 156013, sep 2022.

[25] A. V. Kalueff, M. Gebhardt, A. M. Stewart, J. M. Cachat, M. Brimmer, J. S. Chawla, C. Craddock, E. J. Kyzar, A. Roth, S. Landsman, S. Gaikwad, K. Robinson, E. Baatrup, K. Tierney, A. Shamchuk, W. Norton, N. Miller, T. Nicolson, O. Braubach, C. P. Gilman, J. Pittman, D. B. Rosemberg, R. Gerlai, D. Echevarria, E. Lamb, S. C. F. Neuhauss, W. Weng, L. Bally-Cuif, H. Schneider, and t. Z. Neuros, "Towards a comprehensive catalog of zebrafish behavior 1.0 and beyond," *Zebrafish*, vol. 10, no. 1, pp. 70–86, Mar. 2013.

[26] A. V. Kalueff, A. M. Stewart, and R. Gerlai, "Zebrafish as an emerging model for studying complex brain disorders," *Trends in Pharmacological Sciences*, vol. 35, no. 2, pp. 63–75, Feb. 2014.

[27] G. Kogut and M. Trivedi, "Maintaining the identity of multiple vehicles as they travel through a video network," in *Proceedings IEEE Workshop on Multi-Object Tracking*, 2001, pp. 29–34.

[28] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[29] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking," *arXiv:1504.01942*, apr 2015.

[30] L. Leal-Taixé, A. Milan, K. Schindler, D. Cremers, I. Reid, and S. Roth, "Tracking the trackers: an analysis of the state of the art in multiple object tracking," *arXiv:1704.02781*, 2017.

[31] C.-Y. Lin, C.-Y. Chiang, and H.-J. Tsai, "Zebrafish and Medaka: new model organisms for modern biomedical research," *Journal of Biomedical Science*, vol. 23, no. 1, Jan. 2016.

[32] Q. Liu, D. Chen, Q. Chu, L. Yuan, B. Liu, L. Zhang, and N. Yu, "Online multi-object tracking with unsupervised re-identification learning and occlusion estimation," *Neurocomputing*, vol. 483, pp. 333–347, Apr. 2022.

[33] X. Liu, Y. Yue, M. Shi, and Z.-M. Qian, "3-D video tracking of multiple fish in a water tank," *IEEE Access*, vol. 7, pp. 145 049–145 059, 2019.

[34] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *International Journal of Computer Vision (IJCV)*, vol. 129, no. 2, p. 548–578, oct 2021.

[35] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial Intelligence*, vol. 293, p. 103448, 2021.

[36] S. Macrí, D. Neri, T. Ruberto, V. Mwaffo, S. Butail, and M. Porfiri, "Three-dimensional scoring of zebrafish behavior unveils biological phenomena hidden by two-dimensional analyses," *Nature*, vol. 7, no. 1, may 2017.

[37] T. Meinhardt, A. Kirillov, L. Leal-Taixé, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 8844–8854.

[38] D. A. Meshalkina, M. N. Kizlyk, E. V. Kysil, A. D. Collier, D. J. Echevarria, M. S. Abreu, L. J. G. Barcellos, C. Song, J. E. Warnick, E. J. Kyzar, and A. V. Kalueff, "Zebrafish models of autism spectrum disorder," *Experimental Neurology*, vol. 299, pp. 207–216, Jan. 2018.

[39] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv:1603.00831*, mar 2016.

[40] N. Miller and R. Gerlai, "Quantification of shoaling behaviour in zebrafish (Danio rerio)," *Behavioural Brain Research*, vol. 184, no. 2, pp. 157–166, Dec. 2007.

[41] H. J. Mönck, A. Jörg, T. v. Falkenhausen, J. Tanke, B. Wild, D. Dormagen, J. Piotrowski, C. Winklmayr, D. Bierbach, and T. Landgraf, "BioTracker: an open-source computer vision framework for visual animal tracking," *arXiv:1803.07985*, 2018.

[42] K. Müller, J. Schlemper, L. Kuhnert, and K. D. Kuhnert, "Calibration and 3D ground truth data generation with orthogonal camera-setup and refraction compensation for aquaria in real-time," in *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, vol. 3, Jan. 2014, pp. 626–634.

[43] S. Ohayon, O. Avni, A. L. Taylor, P. Perona, and S. E. R. Egnor, "Automated multi-day tracking of marked mice for the analysis of social behaviour," *Journal of Neuroscience Methods*, vol. 219, no. 1, pp. 10 – 19, 2013.

[44] M. Pedersen, J. B. Haurum, P. Dendorfer, and T. B. Moeslund, "MOTCOM: The multi-object tracking dataset complexity metric," in *Computer Vision – ECCV 2022*. Springer, 2022, pp. 20–37.

[45] M. Pedersen, J. B. Haurum, S. Hein Bengtson, and T. B. Moeslund, "3d-zef: A 3d zebrafish tracking benchmark dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2423–2433.

[46] M. Pedersen, S. Hein Bengtson, R. Gade, N. Madsen, and T. B. Moeslund, "Camera calibration for underwater 3D reconstruction based on ray tracing using Snell's law," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2018.

[47] A. Pérez-Escudero, J. Vicente-Page, R. C. Hinz, S. Arganda, and G. G. De Polavieja, "idtracker: tracking individuals in a group by automatic identification of unmarked animals," *Nature methods*, vol. 11, no. 7, pp. 743–748, 2014.

[48] Z.-M. Qian and Y. Q. Chen, "Feature point based 3D tracking of multiple fish from multi-view images," *PLOS ONE*, vol. 12, no. 6, pp. 1–18, 2017.

[49] Z.-M. Qian, X. E. Cheng, and Y. Q. Chen, "Automatically detect and track multiple fish swimming in shallow water with frequent occlusion," *PLOS ONE*, vol. 9, no. 9, pp. 1–12, 2014.

[50] Z.-M. Qian, S. H. Wang, X. E. Cheng, and Y. Q. Chen, "An effective and robust method for tracking multiple fish in video image based on fish head detection," *BMC Bioinformatics*, vol. 17, no. 1, p. 251, Jun. 2016.

[51] Z. Qian, M. Shi, M. Wang, and T. Cun, "Skeleton-based 3D tracking of multiple fish from two orthogonal views," in *Communications in Computer and Information Science*. Springer Singapore, 2017, pp. 25–36.

[52] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[53] B. Risse, D. Berh, N. Otto, C. Klämbt, and X. Jiang, "FIMTrack: An open source tracking and locomotion analysis software for small animals," *PLOS Computational Biology*, vol. 13, no. 5, pp. 1–15, 2017.

[54] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Computer Vision – ECCV 2016 Workshops*. Springer, 2016, pp. 17–35.

[55] A. Rodriguez, H. Zhang, J. Klaminder, T. Brodin, P. L. Andersson, and M. Andersson, "ToxTrac: a fast and robust software for tracking organisms," *Methods in Ecology and Evolution*, vol. 9, no. 3, pp. 460–464, Sep. 2017.

[56] F. Romero-Ferrero, M. G. Bergomi, R. C. Hinz, F. J. H. Heras, and G. G. d. Polavieja, "idtracker.ai: tracking all individuals in small or large collectives of unmarked animals," *Nature Methods*, vol. 16, no. 2, pp. 179–182, jan 2019.

[57] E. Simetti, F. Wanderlingh, S. Torelli, M. Bibuli, A. Odetti, G. Bruzzone, D. L. Rizzini, J. Aleotti, G. Palli, L. Moriello, and U. Scarcia, "Autonomous underwater intervention: Experimental results of the maris project," *IEEE Journal of Oceanic Engineering*, vol. PP, no. 99, pp. 1–20, 2017.

[58] V. H. Sridhar, D. G. Roche, and S. Gingins, "Tracktor: Image-based automated tracking of animal movement and behaviour," *Methods in Ecology and Evolution*, vol. 10, no. 6, pp. 815–820, Mar. 2019.

[59] A. M. Stewart, F. Grieco, R. A. J. Tegelenbosch, E. J. Kyzar, M. Nguyen, A. Kaluyeva, C. Song, L. P. J. J. Noldus, and A. V. Kalueff, "A novel 3D method of locomotor analysis in adult zebrafish: Implications for automated detection of CNS drug-evoked phenotypes," *Journal of Neuroscience Methods*, vol. 255, pp. 66–74, Nov. 2015.

[60] R. Stiefelhagen, K. Bernardin, R. Bowers, J. Garofolo, D. Mostefa, and P. Soundararajan, "The clear 2006 evaluation," in *Multimodal Technologies for Perception of Humans*. Berlin, Heidelberg: Springer, 2007, pp. 1–44.

[61] R. Sundararaman, C. De Almeida Braga, E. Marchand, and J. Pettre, "Tracking pedestrian heads in dense crowd," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 3865–3875.

[62] V. Takala and M. Pietikainen, "Multi-object tracking using color, texture and motion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2007.

[63] T. Treibitz, Y. Schechner, C. Kunz, and H. Singh, "Flat refractive geometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 34, no. 1, pp. 51–65, jan 2012.

[64] D. Tuia, B. Kellenberger, S. Beery, B. R. Costelloe, S. Zuffi, B. Risse, A. Mathis, M. W. Mathis, F. van Langevelde, T. Burghardt, R. Kays, H. Klinck, M. Wikelski, I. D. Couzin, G. van Horn, M. C. Crofoot, C. V. Stewart, and T. Berger-Wolf, "Perspectives in machine learning for wildlife conservation," *Nature Communications*, vol. 13, no. 1, feb 2022.

[65] S. V. Viscido, J. K. Parrish, and D. Grünbaum, "Individual behavior and emergent properties of fish schools: a comparison of observation and theory," *Marine Ecology Progress Series*, vol. 273, pp. 239–249, 2004.

[66] J. Wall, G. Wittemyer, B. Klinkenberg, and I. Douglas-Hamilton, "Novel opportunities for wildlife conservation and research with real-time monitoring," *Ecological Applications*, vol. 24, no. 4, pp. 593–601, jun 2014.

[67] S. H. Wang, J. Zhao, X. Liu, Z. Qian, Y. Liu, and Y. Q. Chen, "3D tracking swimming fish school with learned kinematic model using LSTM network," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 1068–1072.

[68] S. H. Wang, X. Liu, J. Zhao, Y. Liu, and Y. Q. Chen, "3D tracking swimming fish school using a master view tracking first strategy," in *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, Dec. 2016.

[69] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 3645–3649.

[70] Z. Wu, W. Ke, C. Wang, W. Zhang, and Z. Xiong, "Online 3d reconstruction of zebrafish behavioral trajectories within a holistic perspective," in *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, Dec. 2022.

[71] Z. Wu, N. I. Hristov, T. H. Kunz, and M. Betke, "Tracking-reconstruction or reconstruction-tracking? comparison of two multiple hypothesis tracking approaches to interpret 3d object motion from several camera views," in *Workshop on Motion and Video Computing (WMVC)*. IEEE, Dec. 2009.

[72] Y. Xu, Y. Jin, Y. Zhang, Q. Zhu, Y. He, and H. Sheng, "3d zebrafish tracking with topology association," *IET Image Processing*, vol. 17, no. 4, pp. 1044–1059, nov 2022.

[73] A. Yamashita, E. Hayashimoto, T. Kaneko, and Y. Kawata, "3-d measurement of objects in a cylindrical glass water tank with a laser range finder," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, 2003, pp. 1578–1583.

[74] J. Yin, W. Wang, Q. Meng, R. Yang, and J. Shen, "A unified object motion and affinity model for online multi-object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 6767–6776.

[75] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 11 784–11 793.

[76] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020.

[77] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "ByteTrack: Multi-object tracking by associating every detection box," in *Computer Vision – ECCV 2022*. Springer, 2022, pp. 1–21.

[78] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *International Journal of Computer Vision (IJCV)*, vol. 129, no. 11, pp. 3069–3087, Sep. 2021.

[79] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 22, no. 11, pp. 1330–1334, 2000.

[80] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Computer Vision – ECCV 2020*. Springer, 2020, pp. 474–490.

Chapter 3

Monitoring Marine Organisms in the Wild

Having gained knowledge about detecting and tracking fish in a controlled environment, we now take a plunge into the wild. Monitoring organisms in the wild generally presents a greater challenge compared to the controlled environment. This challenge stems from gathering relevant data and developing robust and general methods that can cope with the changing, unpredictable, and wild underwater environments.

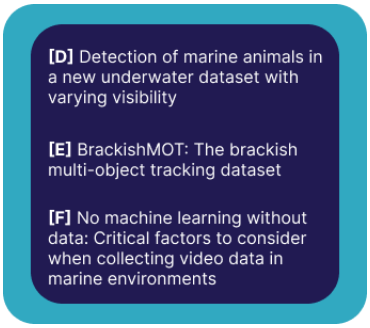


Fig. 3.1: An overview of the papers laying the foundation for the findings in this chapter.

Automating the monitoring task allows marine researchers to scale both data collection and data analysis. Additionally, solutions based on visual sensors can be implemented in a far less intrusive and harmful manner compared to traditional methods which typically involves catching the organisms. There has generally been a lack of relevant underwater computer vision datasets. Up until 2019 when we published the Brackish dataset, described in Paper D, only very few annotated underwater datasets with fish had been made publicly available [7, 13, 18, 21] and they had mainly been captured in clear water and with distinctive fish. A consequence of such data shortage is that all the solutions are developed and assessed on either the same limited set of images or on private datasets [26, 27, 34]. However, during the past years the number of annotated underwater datasets with marine organisms has fortunately increased allowing for better assessment and development of more robust methods [2, 5, 11, 20, 23, 29, 47].

1 Developing an Underwater Object Detection Dataset in Local Brackish Water

The initial steps in capturing any type of image dataset involves selecting a suitable location for recording and designing the camera setup accordingly. However, additional planning is generally required when the location is placed under water. In our case, we aimed to record continuous data from an underwater location in Limfjorden for extended periods due to the requests of local marine researchers. Consequently, we were dependent on either having easy access to the setup in order to change batteries and retrieve data, or place it in proximity to local land-based infrastructure, to facilitate power supply and remote connectivity. With permission from Aalborg Municipality, we were allowed to mount the setup directly on one of the pillars of the Limfjord bridge as illustrated in Figure 3.2. We chose a position nine meters below surface, just above a boulder barrier which we expected to be a diverse habitat for marine life. From this location we could drag cables directly into the bridge house to get a stable power supply and remote access. More details regarding the initial investigative phase of this project is provided in Section 3 of this chapter.

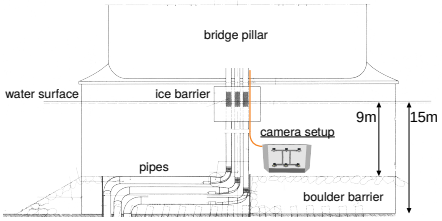


Fig. 3.2: Illustration of the placement of the camera setup on one of the bridge pillars of the Limfjord bridge. The figure is adapted from [37], Paper D, © 2019 IEEE.

The setup consisted of three cameras and three lights mounted on a metal frame as illustrated in Figure 3.3. The frame was designed such that divers could easily change the position and orientation of the cameras and lights according to the prerequisites of various projects. The chosen location for the setup is characterized by strong water currents due to the placement between two bridge pillars in a relatively narrow strait. Therefore, it was imperative to ensure that the setup could withstand the strong current and floating debris. To address this concern, a custom-made metal frame was designed to enclose the cameras and lights, ensuring robustness and stability.

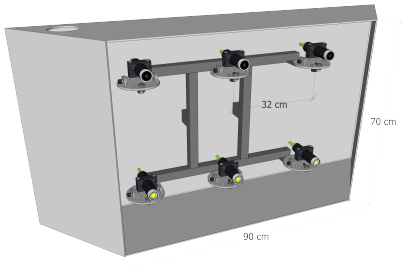


Fig. 3.3: A 3D drawing of the underwater camera setup used to collect the data for the Brackish dataset. The figure is from [37], Paper D, © 2019 IEEE.

After setting up the underwater camera system, data collection was initiated and it continued for two years, resulting in thousands of hours of footage. Particularly in winter, prolonged periods elapsed without any discernible activity within the camera's field of view. Therefore, we implemented a rudimentary approach based on background subtraction utilizing Gaussian Mixture Models to detect any rapid changes within the scene. Given the lack of suitable object detection and classification models, combined with uncertainty regarding the types of organisms to be detected, we stored all sequences in which movement was detected. Subsequently, through a manual selection process we identified 89 diverse sequences encompassing various types of objects, including fish, crabs, starfish, jellyfish, and shrimps. Four frames are presented in Figure 3.4, which shows variation of the dataset with respect to objects and turbidity. The buoy and calibration block in the background were used for other projects during the time of recording and they have not been actively used in the work of this thesis. In total, the sequences contain 14,518 images with 25,613 manually annotated bounding boxes, which laid the foundation of the Brackish dataset presented in Paper D. The dataset was updated with close to 14,000 additional bounding box annotations approximately a year after its release. To this day the Brackish dataset has more than 32,000 views and 2,500 downloads proving its relevance to the field.

Fig. 3.4: Frames from the Brackish dataset [37]. In the top left image is a *big fish* (*lumpsucker*), to the right is a *starfish*, and in the bottom row we see a *jellyfish* to the left and a group of *small fish* (likely *sticklebacks*) to the right. The classes are deliberately coarse due to the difficulty of distinguishing between similarly looking species. The figures are from [37], Paper D, © 2019 IEEE.

Object Detection Baseline Results

At the time when Paper D was published, every computer vision field was experimenting with deep learning models and evaluating their ability to generalize to their specific domain by comparing them with classical approaches [15, 41, 44, 50]. The marine environment was also subject to this trend, and multiple studies demonstrated that convolutional neural networks were superior to classical computer vision and machine learning algorithms [3, 49, 53]. Therefore, to present baseline detection results on the Brackish dataset, we fine-tuned two, at that time, state-of-the-art object detectors, namely YOLOv2 [42] and YOLOv3 [43]. One of the main reasons for choosing these exact models was due to the data they had been trained on, namely: the Open Images dataset [25], which contained relevant classes, and underwater images from the National Oceanic and Atmospheric Administration (NOAA). More specifically:

- *The YOLOv2 detector is obtained from the VIAME toolkit [8] and is pre-trained on ImageNet and fine-tuned on fish datasets from NOAA Fisheries Strategic Initiative on Automated Image Analysis.*
- *The YOLOv3 detector is used in its original version pre-trained on the Open Images dataset [43].*

The YOLOv2 detector is already fine-tuned on underwater images, but contains only the two classes: ***Vertebrates*** *and* ***Invertebrates****. Therefore, further fine-tuning is needed in order to be able to evaluate this model on the proposed dataset. The YOLOv3 detector is trained on the Open Images dataset, which contains 601 classes where five of those are relevant:* ***fish, starfish, jellyfish, shrimp****, and* ***crab****.*
- citation from [37], Paper D, © 2019 IEEE.

Next, the models were fine-tuned on the training sequences of the Brackish dataset with the early layers frozen to maintain the pre-trained and more general feature descriptors. The Brackish dataset was partitioned into three subsets for training, validation, and testing, respectively. A random partitioning scheme was employed, resulting in a split of 80% for training, 10% for validation, and 10% for testing. The models were evaluated by the "AP" which is computed as the average mAP (mean average precision) across the intersection over union (IoU) thresholds [0.50, 0.55, ..., 0.90, 0.95]. Additionally, we also presented the AP_{50} score, which is the AP with an IoU threshold of 0.50.

The main findings are presented in Table 3.1. Note, the only difference between the two sets of categories is that the *fish* class of Open Images is split into *big fish* and *small fish* in the Brackish categories. We see that the YOLOv3 model that has not been fine-tuned does not generalize to the environment of the Brackish dataset, despite having been trained on similar classes. However, opposed to YOLOv2, fine-tuning on the Brackish dataset significantly increases performance for YOLOv3.

Model	Categories	AP	AP_{50}
YOLOv3	Open Images	0.0022	0.0035
YOLOv2 fine-tuned	Open Images	0.0748	0.2577
YOLOv3 fine-tuned	Open Images	**0.3947**	**0.8458**
YOLOv2 fine-tuned	Brackish	0.0984	0.3110
YOLOv3 fine-tuned	Brackish	**0.3893**	**0.8372**

Table 3.1: Performance results of the object detection models on the test split of the Brackish dataset. The table is adapted from [37], Paper D, © 2019 IEEE.

The reason for the low performance of YOLOv2 is likely twofold. The YOLOv2 model has a shallower classification network, Darknet-19 [42], whereas YOLOv3 is based on Darknet-53 [43], which is deeper and also includes residual connections [15]. Additionally, it is not unlikely that the fine-tuning on the NOAA fish data is, ironically, harmful for the performance. The model may have been overfitted to their data, which is possibly significantly different compared to the Brackish dataset, and thereby provides a poor starting point for the fine-tuning.

In Table 3.2 we take a look into the per-class object detection performance of the YOLOv3 model fine-tuned on the Brackish categories. Generally, the performance is satisfactory, but there is room for improvement. However, we see that the performance is particularly high for the *crab* and *starfish* despite they are typically the smallest and most camouflaged objects. This is likely due to the objects often staying in the same place for longer periods of time combined with the random composition of the dataset splits. In other words, frames from the same sequence may be included in both the train and test splits and this is likely the cause of the model to some degree overfitting to the data, and especially the slow moving objects.

Class	AP	AP_{50}
Big fish	0.4621	0.8999
Crab	0.4205	0.9271
Jellyfish	0.3746	0.8205
Shrimp	0.3238	0.7662
Small fish	0.2449	0.6229
Starfish	0.5102	0.9867

Table 3.2: The per-class performance of the YOLOv3 model fine-tuned on the Brackish dataset. The table is adapted from [37], Paper D, © 2019 IEEE.

Despite possible overfitting, we find that it is feasible for object detectors to learn to recognise six common organisms found in Limfjorden. This was not a matter of course, as the Brackish dataset was the first of its kind recorded under such turbid and challenging conditions. While detecting objects can be a challenging task on its own, tracking multiple objects with similar appearances presents an even greater challenge. Therefore, in Paper E, we extended the Brackish dataset by including tracks for each individual object. This work will be described in further details in the next section.

2 Wild and Synthetic: An Exploration of Underwater Multi-object Tracking

Unlike object detection, multi-object tracking in marine environments has not received the same level of attention in recent years. To the best of our knowledge, there exists no annotated and publicly available underwater MOT dataset captured in non-tropical regions and the general lack of MOT datasets has been expressed as a concern in the field [6, 34]. Currently, the only underwater MOT dataset with publicly available annotations is the Fish4Knowledge (F4K) dataset [13], which was captured off the coast of Taiwan in mostly clear waters a decade ago. For obvious reasons, the number of dedicated underwater multi-object trackers is therefore also on the low side [13, 19, 32, 34]. To address this issue, in Paper E we propose to expand the Brackish dataset [37] with nine additional sequences and annotations following MOTChallenge annotation standard [9]. Additionally, we propose a framework for generating synthetic data, which has recently gained attention within other domains [1, 12, 24]. There has also been limited work on synthesizing underwater images to account for the lack of annotated marine data, but mainly for object detection [31, 36].

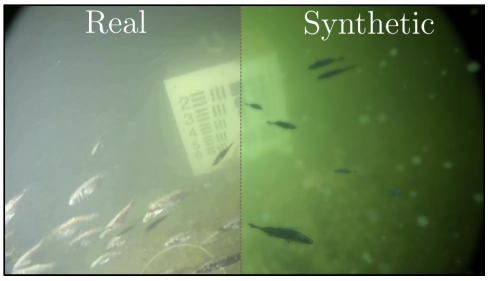


Fig. 3.5: An illustrative portrayal of a sample from a real and synthetic sequence of BrackishMOT [39].

Regarding underwater MOT, Martija and Naval [34] proposed a method for synthesizing sequences based on segmenting objects and subsequently project the segmented fish into new scenes. To emulate motion, the segmentations were temporally rotated and translated. This is a simplified approach, which is not suitable for large-scale data generation. Therefore, we take it a step further and propose a novel framework to synthesize realistic and varied underwater MOT sequences using a rigged 3D fish model and by simulating various degrees of turbidity, particles in the water, using different background sequences, and simulating fish schooling behavior using boids [14].

In the next section we will dive into the BrackishMOT specifications and describe how MOTCOM [38] from Paper C was used to design the dataset splits. This is followed by a description of the framework for constructing realistic synthetic underwater MOT sequences.

Clear as Water: A Balanced and Transparent Dataset Composition

The BrackishMOT dataset contains six distinct categories, namely: *fish*, *crab*, *shrimp*, *starfish*, *small fish*, and *jellyfish*. However, the distribution of sequences, frames, and bounding boxes across the classes is not evenly balanced as illustrated in Figure 3.6. Additionally, the appearance, behavior, and number of objects also vary greatly between the classes. The *crabs* typically move slowly across the sea floor and the *starfish* seldom moves at all. The *small fish* tend to appear in groups, whereas the *fish* exhibits a preference for solitude. Finally, the *jellyfish* gracefully floats with the current, and while the *shrimps* generally partake in this behavior, bursts of acceleration are displayed.

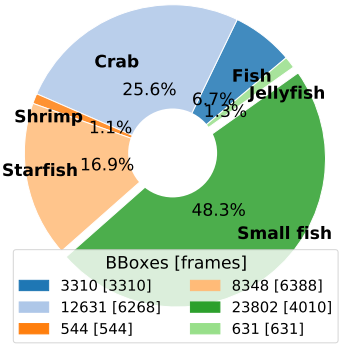
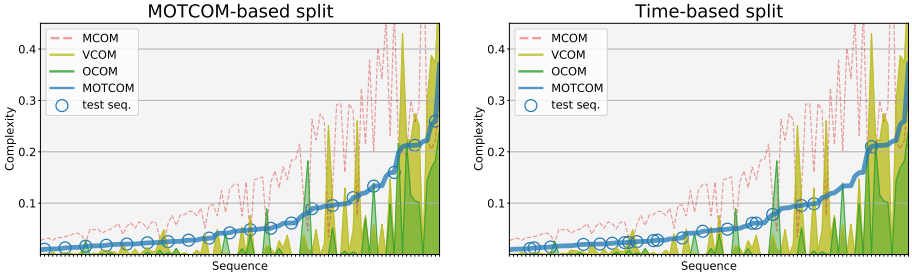


Fig. 3.6: *"Class distribution of the brackishMOT dataset based on the number of bounding boxes."* [39], figure from [39], Paper E.

The in-balance in class distribution and object behavior necessitate careful division of the sequences in the dataset in order to create balanced train and test splits. Therefore, we sort the sequences according to their MOTCOM scores [38], Paper C, and follow a procedure where we assign the most complex sequence to the train split and the second most complex sequence to the test split. Next, we allocate every fifth sequence to the test split, while the remaining sequences are included in the train split. This gives a total of 78 train and 20 test sequences with a balanced distribution between the two splits with respect to the MOT complexity as illustrated in Figure 3.7a. Alternatively, had we selected the first 20 recorded sequences for the test split, the distribution would be skewed with respect to complexity, see Figure 3.7b.



(a) *"Sorting the sequences based on MOTCOM and taking every fifth to be included in the test split. This is the approach we follow."* [39].

(b) *"A typical test split consisting of the 20 first recorded sequences. This approach clearly skews the splits with respect to complexity."* [39].

Fig. 3.7: *"These plots illustrate MOTCOM and the sub-metrics for all the BrackishMOT sequences. In both plots, the sequences are sorted based on their MOTCOM score. The circles mark the test sequences with respect to the split-scheme."* [39], figures are from [39], Paper E.

Swimming in Data: A Framework for Generating Synthetic Underwater Multi-object Tracking Sequences with Fish

Beside proposing the BrackishMOT dataset in Paper E, we also presented a novel framework for generating synthetic underwater MOT sequences with fish. The aim with the framework was three-fold: 1) investigate whether it is possible to detect and track fish in the wild with a model that has been fine-tuned purely on synthetic data, 2) determine whether synthetic data can increase tracker performance on the BrackishMOT test split, and 3) make the framework general in the sense that it can be easily adjusted to fit other underwater environments. The latter was especially critical as a framework that solely expands the BrackishMOT dataset with synthetic sequences would not be beneficial for the broader community.

The proposed framework contains three elements: a fish model, a motion model, and the surrounding environment. The most interesting object in perspective of MOT is the *small fish* class as it often appears in groups that are socially active, leading to sequences with visually similar objects, frequent occlusions, and occasional erratic motion. For this reason, we deliberately focused on the *small fish* class and chose a model that resembles a *stickleback*, which is the most typical family of the *small fish* class. The 3D model was acquired from the fish database of images and photogrammetry [22] and the mesh of the model can be seen in Figure 3.8.

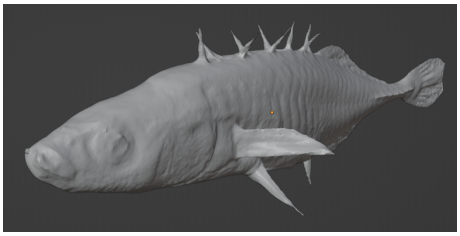


Fig. 3.8: High resolution mesh of the *stickleback* model used in our framework. The figure is from [39], Paper E.

The *small fish* are often seen in groups in the BrackishMOT sequences. Therefore, it was essential to find a motion model that takes group behavior into account. Such a model was proposed in the late 80's by Craig Reynolds [45] when he introduced boids for simulating flock behavior of birds. Boid behavior is based on each individual object following simple rules that determines how they should act according to their neighbors. Our implementation of boid behavior involves four forces that are applied to the direction and velocity of every object, namely: separation, cohesion, alignment, and leader. The forces are dependent on the proximity and behavior of the closest neighbors and ensure that the fish stay close, avoid total crowding, and to some degree follow a leader object. The interaction between the objects causes this simple set of rules to simulate relatively complex social behavior. Additionally, by adjusting the weight of the individual forces in a stochastic manner it is possible to obtain more complex and varied motion.

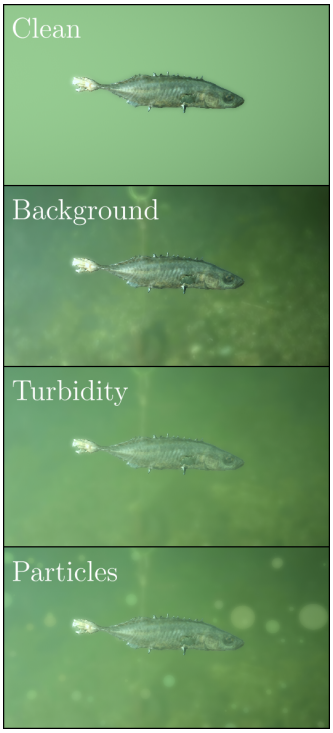


Fig. 3.9: The stickleback model is displayed on a plain background in the top image and in the image below we have introduced a background video without real-life objects. To make the environment more realistic we add artificial turbidity, and lastly, floating particles are introduced to simulate noise to the environment.

The habitat and the surrounding conditions also have a major effect on the general complexity of underwater scenes. Therefore, we propose a practical method for simulating varied and changing environments. We base our initial habitat model on the conditions observed in Limfjorden, which are characterized by a significant variation in the amount of suspended sediment, water current strength, salinity, and other factors. All of these contribute to changes in the amount of light, the perceived color of the water, visibility, and number of floating particles. We propose three steps to create a realistic looking underwater environment and the steps are visualized in Figure 3.9. First, we introduce a background video, in our case we have used videos from the Brackish setup where there are no organisms present. We then add turbidity, which is implemented as a spherical "fog" that intensifies with the distance to the camera. Lastly, we add artificial floating particles which are represented as blurry spheres.

Through the utilization of diverse background videos, adjustments in turbidity-fog levels and color, as well as modifications in the size, color, and quantity of floating particles, a wide spectrum of diverse environments can be generated. This, in conjunction with the possibility to modify the object count and behavioral characteristics, facilitates effortless creation of an extensive and varied synthetic dataset resembling real-world data captured in the wild.

Blending Real and Synthetic Data to Advance Fish Tracking in the Wild

We provide baseline results for the BrackishMOT dataset by fine-tuning the state-of-the-art tracker CenterTrack [54] (CT) following different training strategies involving: real-data, synthetic data, and combinations of the two. The authors of CenterTrack have published two pre-trained models trained on MS COCO [28] and ImageNet [10] and a recipe for fine-tuning their models, which we use as our basis for the evaluation. We assess the performance of our models by the conventional MOT metrics MOTA [4], IDF1 [46], and HOTA [30]. In total we evaluate six models and the results are presented in

Model	HOTA↑	MOTA↑	IDF1↑
CT-COCO-Brack	0.36	0.37	0.39
CT-COCO-Synth	0.36	0.38	0.39
CT-COCO-Mix	0.36	0.37	0.39
CT-ImNet-Brack	0.38	0.43	0.44
CT-ImNet-Synth	0.38	0.42	0.41
CT-ImNet-Mix	**0.40**	**0.44**	**0.45**

Table 3.3: Baseline results for the six CenterTrack (CT) models evaluated on the BrackishMOT test split. The table is adapted from [39], Paper E.

Table 3.3. The model name indicates the training strategy, e.g., CT-COCO-Brack has been pre-trained on MS COCO and fine-tuned on BrackishMOT. The CT-COCO-Synth and CT-ImNet-Synth models have been fine-tuned on the synthetic sequences and subsequently fine-tuned on BrackishMOT. Lastly, CT-COCO-Mix and CT-ImNet-Mix have been fine-tuned on a combination of the synthetic and real sequences in one go.

Based on the findings, it is evident that a CenterTrack model pre-trained on ImageNet is the better option compared to MS COCO when the model is subsequently fine-tuned on the BrackishMOT dataset following the proposed strategies. Additionally, we find that the synthetic sequences, despite only containing a model resembling the *small fish* class, can increase the performance of CenterTrack. However, it is also apparent that the increase is marginal, and that there is room for improvement.

The synthetic environment only contains objects of the *small fish* class, therefore, we also investigated how a model purely fine-tuned on the synthetic data would perform on a subset of the BrackishMOT test split consisting only of the eight *small fish* sequences. We name these models CT-ImNet-Synthetic and CT-COCO-Synthetic and it should be stressed that they have **not** been fine-tuned on real underwater data. The results are presented in Table 3.4. Interestingly, we see that the MS COCO pre-trained model is better suited for this task compared to ImageNet. The performances are not striking for any of the two models, but nonetheless, they indicate that it is possible to teach models to track marine organisms in challenging environments from purely synthetic data (if we ignore pre-training).

Model	HOTA↑	MOTA↑	IDF1↑
CT-ImNet-Synthetic	0.17	0.13	0.19
CT-COCO-Synthetic	**0.21**	**0.18**	**0.24**
CT-ImNet-Brack	0.39	0.50	0.46
CT-COCO-Brack	0.37	0.47	0.43

Table 3.4: The models have been evaluated on a subset of the BrackishMOT test split, which only consists of the sequences containing the *small fish* class. The CT-ImNet-Synthetic and CT-COCO-Synthetic models have only been fine-tuned on the synthetic sequences. The table is adapted from [39], Paper E.

To conclude, we find that it is possible to achieve decent performance on the challenging multi-object tracking sequences of the BrackishMOT dataset with a tracker developed mainly for terrestrial tasks. Incorporating synthetic data during training can marginally improve performance of the chosen tracker, but our experiments suggest that there is potential for more.

3 Underwater Image Acquisition: Lessons Learned

During the design and development phase of the underwater camera and lighting system used to capture the Brackish dataset, we were unable to locate a practical overview of factors to be aware of when establishing a setup focused on monitoring marine life in the wild. There exists great literature on topics such as the variability of ocean color [35], the impact of eutrophication on coastal turbidity [51], and techniques to mitigate backscatter [17, 33]. However, in a field characterized by a substantial scarcity of data, it is an uphill battle to expect marine researchers, some without a technical background, to conduct a literature review ahead of potentially designing a camera system. For this reason, we wrote an essay about what we consider to be essential factors to contemplate for optimizing visual data acquisition in underwater environments and it is based on the experience and knowledge gained in the process of producing the Brackish datasets. The essay is presented in Paper F and it is written in an easily digestible manner to a cross-disciplinary audience with limited technical knowledge with the aim of demystifying underwater image acquisition and encouraging marine researchers to utilize visual data and computer vision to optimize data analysis. The basis of the essay revolves around the following six elements: 1) attenuation of light, 2) backscatter, 3) artificial light, 4) refraction, 5) data handling, and 6) the local environment. Our knowledge regarding this topic comes from the development of the camera setup presented in Paper D, where we had to consider each of the elements.

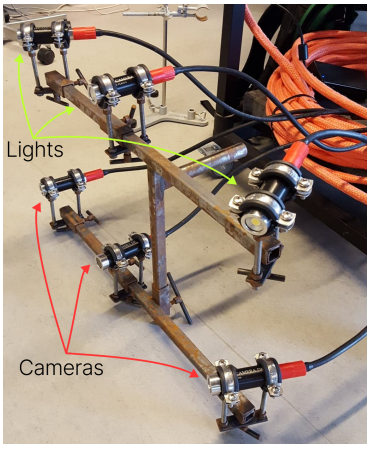


Fig. 3.10: The prototype setup used for initial investigations with respect to location and orientation of the camera and lights.

The initial intention with the camera setup was to acquire practical know-how in using cameras and lights in Limfjorden, with the goal of preparing for a future iteration of the setup that would be portable. This involved gathering insights throughout the years regarding the fluctuating turbidity levels, algae bloom on the lenses, optimal positioning of lights and cameras, and, naturally, capturing sequences of organisms. Limfjorden is a strait that runs through the northern part of Jutland from the North Sea in the west to Kattegat in the east and it is of particular interest to local marine biologists as it has witnessed oxygen depletion, severe decline in fish populations, and more over the past decades [16]. The water temperature in Limfjorden can fluctuate by up to twenty degrees, influ-

enced by seasonal variations. Moreover, the water in the strait is brackish due to the inflow of freshwater streams and drainage channels. The latter being a cause of turbidity through eutrophication [51] which can have a strong effect on the attenuation of light, especially during warmer periods. For these reasons, we anticipated that the visibility could be poor and vary to a large degree based on the water current, temperature, depth, and location. Artificial lights can be used to counteract low visibility, if positioned accordingly, and otherwise be the cause of backscatter and shadows [48, 52]. Therefore, we decided to install multiple artificial lights that could be turned on and off depending on the situation. In order to determine the initial position and orientation of the cameras and lights we designed a prototype setup, which is shown in Figure 3.10. We conducted several field tests, see Figure 3.11, with the prototype to find the optimal configuration of the lights and cameras to meet our needs and to reduce the effect of backscatter and attenuation of light. However, our findings revealed that there was no single optimal solution, as it heavily relied on the characteristics of the surrounding environment and the task being performed. For example, images of the sea floor were, intuitively, of higher quality when the cameras were placed in the bottom row of the rack. Additionally, the position of the cameras and the surroundings dictated how and where the lights should be placed and oriented.

Fig. 3.11: Evening experiments with the prototype setup in a shallow location of Limfjorden during Summer.

Fig. 3.12: This is me preparing for an underwater survey near the pillars of the Limfjord bridge.

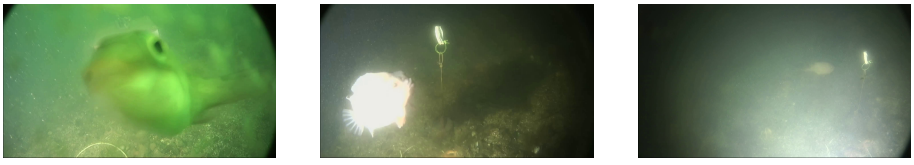
We surveyed positions along the harbor front of Aalborg and had divers do inspections around the bridge pillars of the Limfjord bridge to find a suitable location. See Figure 3.12 for a photo of me preparing for one of the surveys; unfortunately, I was not the one doing the dive as only professional divers are allowed in the water near Aalborg harbor and the Limfjord bridge. Figure 3.13 shows an image from one of the underwater surveys directly below the bridge tower, and yes, the diver is scratching the crab on its back with his gloved hand. Ultimately, we selected this specific location based on several factors. It offered convenient access to a reliable power supply and internet connection. Additionally,

the positioning of an ice barrier directly above the camera setup provided an ideal opportunity to secure the cables and shield them from potential damage caused by floating debris and ice, see Figure 3.2 for an illustration of the ice barrier and camera setup.

Despite a careful design and precautions, we still encountered a range of difficulties when capturing the data due to algae bloom, backscatter, overexposure, shadows, and more. In Figure 3.14a the image has a greenish tone caused by algae bloom on the lense. The fish is a goldsinny wrasse which may have used the setup as its residence for a period of time. In Figure 3.14b a lumpsucker is swimming in front of a light causing its body to reflect the light and be heavily overexposed. Additionally, due to only a single light being turned on, the fish casts a shadow behind it, potentially hiding other objects. Lastly, in Figure 3.14c the scene is hidden behind a bright fog made of small particles that are scattering and reflecting the artificial light. This phenomenon occurs when the camera and lights are closely positioned and the water is turbid.

Fig. 3.13: A diver is investigating the area near a bridge pillar of the Limfjord bridge and has found a crab that is getting a scratch on its back.

It is demanding to capture high quality image data in underwater environments as it is not straightforward to adjust a setup, change a lens, or conduct maintenance of a system while it is submerged under water. However, it is crucial that computer vision researchers get access to larger and more varied underwater datasets, as data is the oxygen of machine learning. Therefore, it is essential that marine researchers gather and publish the best possible data given the existing constraints. To prevent researchers interested in capturing underwater visual data from ending up in deep water, we described the most critical factors to consider, based on our experiences, in Paper F.



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

Fig. 3.14: Three images that illustrates some of the difficulties encountered during recording of data in Limfjorden. a) early algae bloom makes the scene look green. b) overexposure of a fish that swims in front of a light. c) backscatter makes the scene seem foggy and more turbid than it really is. The images are from [40], Paper F.

4 Summary and Scientific Contributions

In this chapter we dived into underwater image acquisition in the wild with a focus on object detection and tracking in brackish waters. The basis of the work conducted in this chapter was the development of the Brackish dataset presented in Paper D. The Brackish dataset was the first bounding box annotated and publicly available underwater object detection dataset of its kind. It consists of 89 sequences with varying visibility and six coarse object categories including fish, crabs, and starfish. Beside publishing the dataset we also presented baseline results from two state-of-the-art object detectors.

Next, we presented the BrackishMOT dataset from Paper E which is a multi-object tracking expansion of the Brackish dataset. BrackishMOT includes nine new sequences, multi-object tracking annotations, and a framework for generating realistic looking synthetic sequences with multiple fish, custom backgrounds, and varying levels of turbidity. Additionally, we investigated multiple training strategies with and without synthetic data and presented preliminary tracking results.

Finally, we provided supplementary details and perspectives on the development of the camera system and data acquisition process. The insights were compiled into a set of factors to consider when recording data in marine environments as outlined in Paper F. We concluded by encouraging marine researchers to consider using cameras and computer vision as a scalable and non-intrusive underwater monitoring solution. Moreover, due to the scarcity of annotated underwater data, we urge researchers to publish recordings and annotations as a means of advancing the field of underwater computer vision.

The main scientific contributions in this chapter can be summarized as:

- In Paper D we presented the first publicly available underwater object detection dataset captured in brackish waters with varying visibility. Additionally, we fine-tuned two state-of-the-art object detectors and provided baseline results.
- In Paper E the first multi-object tracking dataset captured in non-tropical waters was presented along with a framework for creating realistic synthetic sequences of multiple fish. Furthermore, we investigated strategies for training a tracker via combinations of real and synthetic data and presented preliminary baseline results.
- In Paper F we compiled the knowledge gathered through the development of the aforementioned datasets and presented essential factors to consider when recording video data in marine environments.

Future Work

One of the major goals with the underwater camera setup was to obtain knowledge about the rough environment in Limfjorden and how it physically affected the setup over time. This should pave the way for the next and portable iteration of the setup, which ideally can be used for both stationary and moving tasks. For example, recording a limited area of an artificial stone reef or eelgrass plantation for a period of time to analyse growth and life. Or to survey larger areas by mounting the camera on a boat, kayak, or handheld while diving. It is also relevant to capture distance data, e.g., to estimate the size of plants and animals. The preferable solution would be a passive sensor such as a stereo setup, as it does not affect the surroundings as opposed to, e.g., time-of-flight, structured light, and sonar.

Expanding the Brackish dataset with more video sequences could be a way to ensure that it stays relevant to the community. Another solution would be to include additional annotations in the form of, e.g., segmentation masks and provide guidance to create fair data splits, that does not include data leakage where test and train images can be consecutive frames from the same sequence. Lastly, it would be relevant to establish a fine-grained classification branch of the Brackish dataset. This is likely not possible with the current recordings, but a solution could be to record high quality and varied sequences from public aquariums, where they have replicas of natural habitats of straits, estuaries, and stone reefs.

The aforementioned possibilities of expanding the Brackish dataset could with minimal effort lead to an expansion of the BrackishMOT dataset as well. Additionally, there is a huge potential in the synthetic framework for generating varied and relevant underwater multi-object tracking sequences. It would be interesting to include more animal and motion models, e.g., additional fish species, crabs, shrimps, and jellyfish. Moreover, we could implement a force to imitate water current that affects the motion of the animals, particles, and turbidity in the scene.

Currently, the motion model is loosely based on boid-behavior, however, fish are prey animals and they occasionally behave erratically to avoid predators and this type of behavior is currently not directly implemented. In order to make the motion more realistic it may be feasible to include more stochastic variables sporadically affecting the orientation and acceleration of the fish. Another relevant feature would be to simulate the depth at which the synthetic scene is recorded and attenuate the light accordingly. One could also add a feature for simulating weather, e.g., clear or cloudy, day or night and the option of using artificial lights (including shadows and backscatter).

References

[1] G. Amato, L. Ciampi, F. Falchi, C. Gennaro, and N. Messina, "Learning pedestrian detection from virtual worlds," in *Image Analysis and Processing – ICIAP 2019*. Cham: Springer International Publishing, 2019, pp. 302–312.

[2] Australian Institute Of Marine Science, "Ozfish dataset - machine learning dataset for baited remote underwater video stations," 2020, doi: 10.25845/5E28F062C5097.

[3] O. Beijbom, T. Treibitz, D. I. Kline, G. Eyal, A. Khen, B. Neal, Y. Loya, B. G. Mitchell, and D. Kriegman, "Improving automated annotation of benthic survey images using wide-band fluorescence," *Scientific Reports*, vol. 6, no. 1, 2016.

[4] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, pp. 1–10, May 2008.

[5] L. Cai, N. E. McGuire, R. Hanlon, T. A. Mooney, and Y. Girdhar, "Semi-supervised visual tracking of marine animals using autonomous underwater vehicles," *International Journal of Computer Vision (IJCV)*, vol. 131, no. 6, pp. 1406–1427, mar 2023.

[6] M.-C. Chuang, J.-N. Hwang, J.-H. Ye, S.-C. Huang, and K. Williams, "Underwater fish tracking for moving cameras based on deformable multiple kernels," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–11, 2016.

[7] G. Cutter, K. Stierhoff, and J. Zeng, "Automated detection of rockfish in unconstrained underwater videos using haar cascades and a new image dataset: Labeled fishes in the wild," in *Proceedings of the IEEE Winter Applications and Computer Vision Workshops (WACVW)*. IEEE, jan 2015.

[8] M. Dawkins, L. Sherrill, K. Fieldhouse, A. Hoogs, B. Richards, D. Zhang, L. Prasad, K. Williams, N. Lauffenburger, and G. Wang, "An open-source platform for underwater image and video analytics," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017.

[9] P. Dendorfer, A. Osep, A. Milan, K. Schindler, D. Cremers, I. Reid, S. Roth, and L. Leal-Taixé, "Motchallenge: A benchmark for single-camera multiple target tracking," *International Journal of Computer Vision (IJCV)*, vol. 129, p. 845–881, 2021.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2009, pp. 248–255.

[11] E. M. Ditria, S. Lopez-Marcano, M. Sievers, E. L. Jinks, C. J. Brown, and R. M. Connolly, "Automating the analysis of fish abundance using object detection: Optimizing animal ecology with deep learning," *Frontiers in Marine Science*, vol. 7, Jun. 2020.

[12] M. Fabbri, G. Brasó, G. Maugeri, O. Cetintas, R. Gasparini, A. Ošep, S. Calderara, L. Leal-Taixe, and R. Cucchiara, "Motsynth: How can synthetic data help pedestrian detection and tracking?" in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 10 829–10 839.

[13] R. B. Fisher, Y.-H. Chen-Burger, D. Giordano, L. Hardman, and F.-P. Lin, *Fish4Knowledge: Collecting and Analyzing Massive Coral Reef Fish Video Data*. Springer International Publishing, 2016, vol. 104.

[14] C. Hartman and B. Beneš, "Autonomous boids," *Computer Animation and Virtual Worlds*, vol. 17, no. 3-4, pp. 199–206, 2006.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016, pp. 770–778.

[16] E. Hoffmann, "Limfjorden - fiskene der forsvandt," *Fisk & Hav*, no. 53, pp. 40–51, 2001. [Online]. Available: https://orbit.dtu.dk/files/279088250/53_2001_Limfjorden_fiskene_der_forsvandt.pdf

[17] J. Jaffe, K. Moore, J. McLean, and M. Strand, "Underwater optical imaging: Status and prospects," *Oceanography*, vol. 14, no. 3, pp. 64–75, 2001.

[18] J. Jäger, M. Simon, J. Denzler, V. Wolff, K. Fricke-Neuderth, and C. Kruschel, "Croatian fish dataset: Fine-grained classification of fish species in their natural habitat," in *Procedings of the Machine Vision of Animals and their Behaviour Workshop 2015*. British Machine Vision Association, 2015.

[19] J. Jäger, V. Wolff, K. Fricke-Neuderth, O. Mothes, and J. Denzler, "Visual fish tracking: Combining a two-stage graph approach with CNN-features," in *OCEANS 2017 - Aberdeen*. IEEE, Jun. 2017.

[20] A. Jansen, D. Walden, S. Walker, and C. Buccella, "A deep learning dataset for underwater object detection of tropical freshwater fish species in northern australia," 2022, doi: 10.5281/ZENODO.7250921.

[21] A. Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, W.-P. Vellinga, J. Champ, R. Planqué, S. Palazzo, and H. Müller, "Lifeclef 2016: Multimedia life species identification challenges," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*. Cham: Springer International Publishing, 2016, pp. 286–310.

[22] Y. Kano, M. S. Adnan, C. Grudpan, J. Grudpan, W. Magtoon, P. Musikasinthorn, Y. Natori, S. Ottomanski, B. Praxaysonbath, K. Phongsa, A. Rangsiruji, K. Shibukawa, Y. Shimatani, N. So, A. Suvarnaraksha, P. Thach, P. N. Thanh, D. D. Tran, K. Utsugi, and T. Yamashita, "An online database on freshwater fish diversity and distribution in mainland southeast asia," *Ichthyological Research*, vol. 60, no. 3, pp. 293–295, Jun. 2013.

[23] J. Kay, P. Kulits, S. Stathatos, S. Deng, E. Young, S. Beery, G. V. Horn, and P. Perona, "The caltech fish counting dataset: A benchmark for multiple-object tracking and counting," in *Computer Vision – ECCV 2022*. Springer, 2022, pp. 290–311.

[24] P. Krahenbuhl, "Free supervision from video games," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2018.

[25] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The open images dataset v4," *International Journal of Computer Vision (IJCV)*, vol. 128, no. 7, pp. 1956–1981, mar 2020.

[26] A. B. Labao and P. C. Naval, "Cascaded deep network systems with linked ensemble components for underwater fish detection in the wild," *Ecological Informatics*, vol. 52, pp. 103–121, 2019.

[27] ——, "Simultaneous localization and segmentation of fish objects using multi-task cnn and dense crf," in *Intelligent Information and Database Systems*. Cham: Springer International Publishing, 2019, pp. 600–612.

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Computer Vision – ECCV 2014*. Cham: Springer, 2014, pp. 740–755.

[29] C. Liu, H. Li, S. Wang, M. Zhu, D. Wang, X. Fan, and Z. Wang, "A dataset and benchmark of underwater object detection for robot picking," in *Proceedings of the IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*. IEEE, jul 2021.

[30] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *International Journal of Computer Vision (IJCV)*, vol. 129, no. 2, p. 548–578, oct 2021.

[31] A. Mahmood, M. Bennamoun, S. An, F. Sohel, F. Boussaid, R. Hovey, and G. Kendrick, "Automatic detection of western rock lobster using synthetic data," *ICES Journal of Marine Science*, vol. 77, no. 4, pp. 1308–1317, Nov. 2019.

[32] T. Mandel, M. Jimenez, E. Risley, T. Nammoto, R. Williams, M. Panoff, M. Ballesteros, and B. Suarez, "Detection confidence driven multi-object tracking to recover reliable tracks from unreliable detections," *Pattern Recognition*, vol. 135, p. 109107, 2023.

[33] P. Mariani, I. Quincoces, K. Haugholt, Y. Chardard, A. Visser, C. Yates, G. Piccinno, G. Reali, P. Risholm, and J. Thielemann, "Range-gated imaging system for underwater monitoring in ocean environment," *Sustainability*, vol. 11, no. 1, p. 162, Dec. 2018.

[34] M. A. M. Martija and P. C. Naval, "SynDHN: Multi-object fish tracker trained on synthetic underwater videos," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*. IEEE, Jan. 2021.

[35] V. Mascarenhas and T. Keck, "Marine optics and ocean color remote sensing," in *YOUMARES 8 – Oceans Across Boundaries: Learning from each other*. Springer International Publishing, 2018, pp. 41–54.

[36] J. Musić, S. Kružić, I. Stančić, and F. Alexandrou, "Detecting underwater sea litter using deep neural networks: An initial study," in *Proceedings of the International Conference on Smart and Sustainable Technologies (SpliTech)*. IEEE, Sep. 2020.

[37] M. Pedersen, J. Bruslund Haurum, R. Gade, and T. B. Moeslund, "Detection of marine animals in a new underwater dataset with varying visibility," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 18–26.

[38] M. Pedersen, J. B. Haurum, P. Dendorfer, and T. B. Moeslund, "MOTCOM: The multi-object tracking dataset complexity metric," in *Computer Vision – ECCV 2022*. Springer, 2022, pp. 20–37.

[39] M. Pedersen, D. Lehotský, I. Nikolov, and T. B. Moeslund, "BrackishMOT: The brackish multi-object tracking dataset," in *Image Analysis. SCIA 2023.* Springer, 2023, pp. 17–33.

[40] M. Pedersen, N. Madsen, and T. B. Moeslund, "No machine learning without data: Critical factors to consider when collecting video data in marine environments," vol. 16, no. 3, 2021. [Online]. Available: https://www.thejot.net/archive-issues/?id=73

[41] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016.

[42] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," *arXiv:1612.08242*, 2016.

[43] ——, "Yolov3: An incremental improvement," *arXiv:1804.02767*, 2018.

[44] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[45] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH '87*. ACM Press, 1987.

[46] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Computer Vision – ECCV 2016 Workshops*. Springer, 2016, pp. 17–35.

[47] A. Saleh, I. H. Laradji, D. A. Konovalov, M. Bradley, D. Vazquez, and M. Sheaves, "A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis," *Scientific Reports*, vol. 10, no. 1, p. 14671, 2020.

[48] M. Sheinin and Y. Y. Schechner, "The next best underwater view," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2016.

[49] S. A. Siddiqui, A. Salman, M. I. Malik, F. Shafait, A. Mian, M. R. Shortis, and E. S. H. and, "Automatic fish species classification in underwater videos: exploiting pre-trained deep neural network models to compensate for limited labelled data," *ICES Journal of Marine Science*, vol. 75, no. 1, pp. 374–389, 2017.

[50] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

[51] V. H. Smith, "Eutrophication of freshwater and coastal marine ecosystems a global problem," *Environmental Science and Pollution Research*, vol. 10, no. 2, pp. 126–139, mar 2003.

[52] T. Treibitz, Y. Schechner, C. Kunz, and H. Singh, "Flat refractive geometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 34, no. 1, pp. 51–65, jan 2012.

[53] S. Villon, M. Chaumont, G. Subsol, S. Villéger, T. Claverie, and D. Mouillot, "Coral reef fish detection and recognition in underwater videos by supervised machine learning: Comparison between deep learning and hog+svm methods," in *Advanced Concepts for Intelligent Vision Systems*. Cham: Springer International Publishing, 2016, pp. 160–171.

[54] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Computer Vision – ECCV 2020*. Springer, 2020, pp. 474–490.

Chapter 4

Long-Term Monitoring of Giant Sunfish

Computer vision and machine learning algorithms have been used for conservation efforts for decades with re-identification as a key application [33, 37]. Traditionally, re-identification systems have employed a computer vision module to analyse images and identify unique features such as scars, coat patterns, or markings on the animals. Subsequently, a machine learning algorithm is employed to quantify and compare the features to those from other images, allowing for re-identifying individual animals [9, 15, 34]. Handcrafted feature point detectors and descriptors like SIFT [19], RootSIFT [2], ORB [29], and SURF [3] have been used to locate and describe visually distinct points on animals [1, 9, 20, 35, 39].

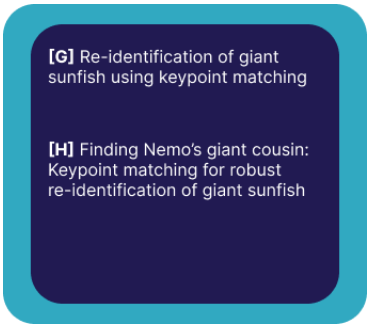


Fig. 4.1: An overview of the papers laying the foundation for the findings in this chapter.

Recently, though, deep neural networks have generally gained increasing attention within the field of animal re-identification [4, 8, 33]. The most common approaches [32] include using triplet loss [6, 17] or designing the re-identification model as a Siamese network [21, 38]. However, re-identification models based on deep learning networks typically require a large amount of training data to ensure that the model generalizes outside the individuals included in the training space. This necessitates that there is a sufficient quantity of data available and that a significant part of the data is annotated. This is often not the case when looking for elusive or rare animals, like the giant sunfish, where the amount of data is scarce.

1 A Unique and Elusive Giant

The giant sunfish (*Mola alexandrini*) is the world's heaviest bony fish [14, 31], but little is generally known about this species. It is typically observed in temperate waters of the Southern Hemisphere and it belongs to the *Molidae* family, which includes the ocean sunfish *Mola mola* that is occasionally found close to Denmark [22]. Apart from being a family of fish that deserves protection and the right to inhabit the Earth on equal terms with all other living organisms, it potentially exerts a substantial influence on the ecosystems in which it resides. The fish belonging to the *Molidae* family primarily consume jellyfish, and, their size taken into consideration, potentially plays a pivotal role in the regulation of these gelatinous organisms, which are blooming worldwide as a consequence of human activities [28]. Not only are sunfish large, goofy, and plain likeable creatures, see Figure 4.2, they are also an effective part of the ocean cleaning staff, that helps maintain healthy ecosystems, which, among many other things, is crucial for large industries like fishing and tourism. Therefore, it is essential that information on these unique creatures is collected and analyzed in order to understand their behavior and life-cycle such that we are able to implement effective conservation efforts where needed.

Fig. 4.2: Four images of giant sunfish from the Match My Mola database [24].

Due to their elusive nature and extensive spatial mobility, sunfish are rarely seen and recorded. Therefore, marine researchers are utilizing citizen science projects to gather images and recordings from amateur divers, tourists, and the likes. Since 2013 the 'Match My Mola' project [24] has received recordings of sunfish, particularly from the Bali area in Indonesia. Bali is a 'hot spot' for sunfish activities and there are near-shore areas where sunfish are regularly seen. Therefore, local dive shops are encouraging tourists to upload their images and videos to the Match My Mola database including location and time. The database is curated by marine biologists investigating whether the same individuals are visiting the same areas recurrently. Additionally, they are exploring issues such as the growth rate of sunfish in the wild, the impact of having boats and tourist divers in close contact with the creatures, and the extent of their habitat range. All of these problems are dependent on being able to identify the same individual repeatedly.

Re-identifying Giant Sunfish

Identifying the same individual on images captured at different times or by different photographers is commonly known as photo-identification by biologists and ecologists, and re-identification in the computer vision community. Typical cues for re-identification of marine animals include pelage patterns [21], caudal and dorsal fins [4, 7], scars, and body markings [16, 23]. However, relying on specific patterns on an organism for re-identification necessitates that these patterns are both unique and persistent. The giant sunfish have such body markings and it has recently been shown that the markings do not significantly change during a span of at least seven years [23].

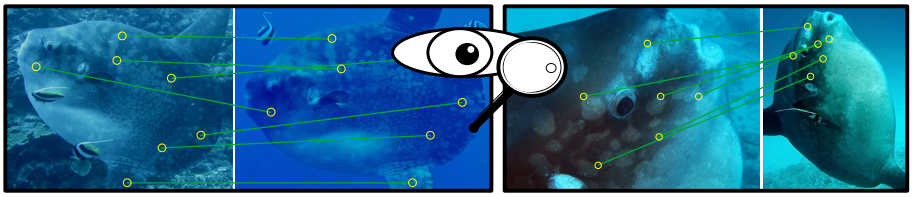


Fig. 4.3: *"Giant sunfish have unique patterns on their bodies, which can be used for photo identification. Traditionally, marine researchers have matched images by manual visual pattern recognition focused on the body markings, as illustrated in these two examples."* [26], the figure is from [26], Paper H.

Currently, marine biologists and trained volunteers are manually re-identifying giant sunfish from the Match My Mola database by comparing their body markings as illustrated in Figure 4.3. The database contains thousands of images making it an extremely time-consuming task that is continuously growing as more images are uploaded to the database. For this reason, in Paper G and Paper H we investigated whether computer vision could be used to automate the re-identification process. At the time when we proposed the method described in Paper G only 29 individuals (and 91 images) had been manually verified to be re-sightings of the same individuals captured either by different divers or at different times. Therefore, we deemed the task unsuitable to solve for deep learning models due to the scarce amount of data and because it made little sense to attempt to create diverse train, validation, and test splits. Instead, we looked for methods that required no training or fine-tuning, such as handcrafted feature descriptors.

The body of the giant sunfish is rigid [10], therefore, the constellation of body markings are mainly subject to translation, scaling, and rotation between sightings. However, there are other factors affecting the clarity and appearance of the patterns like the attenuation of light, the camera and lens, the distance between the diver and the fish, and more. Inspired by traditional image registration techniques we anticipated that keypoint matching could be applicable for automating the task of re-identifying giant sunfish.

2 Connecting the Dots: Keypoint Matching for Re-identifying Giant Sunfish

Detection and description of keypoints have been studied for many years and have been widely used in fields like tracking and image registration. However, we are, of course, not the first to utilize keypoints in the field of re-identification. Keypoint matching has been used for re-identifying patterned species for more than a decade in various constellations [5, 9, 11, 13, 36] mainly based on the scale invariant feature transform (SIFT) algorithm [18, 19]. The pipeline that we proposed in Paper G and expanded upon in Paper H is, therefore, not groundbreaking. However, our approach to the problem diverge from previous practices in the field. Given the severely limited number of manually annotated and verified re-identifications in the Match My Mola database, we designed the solution under the assumption that no training data was available, as this is often the case for conservation and research efforts targeted at specific species. Furthermore, in addition to performance, practicality held significant importance in our considerations.

A Super and On-Point Re-identification Approach

We designed the solution specifically for marine biologists who typically do not have a technical background in computer vision and image analysis. Therefore, we adopted a paradigm stating that the method could only consist of public and widely used algorithms configured with the default parameters, such that it would require no parameter-tuning. The pipeline proposed in Paper G consisted of the modules presented in Figure 4.4, namely: detect object, find keypoints, match keypoints, clean matches, and rank.

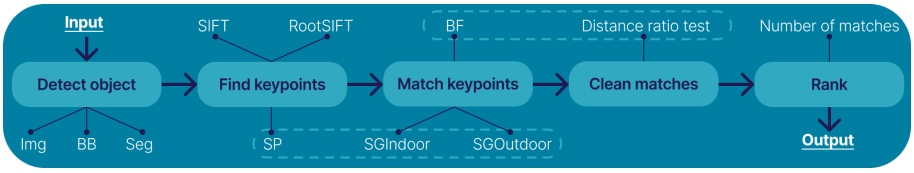


Fig. 4.4: Visual representation of the pipeline presented in Paper G. The components linked to the modules denote variations of the system, e.g., one model is based on segmentation (Seg), SIFT keypoints, and brute force (BF) matching. The dashed lines indicate that only SuperPoint (SP) is matched using SuperGlue (SG) and only the BF matches are cleaned using the distance ratio test.

The components linked to the modules represent choices for configuring the pipeline. We construct and evaluate models for each of the combinations. For example, in the first module we choose between the original image (Img), the object bounding box (BB), or the segmented object (Seg), see Figure 4.5 for

an example of the three levels of segmentation. Note, for each model only one component is used per module.

(a) Original image (img). **(b)** Bounding box (BB). **(c)** Segmentation (Seg).

Fig. 4.5: In the *detect object* module there are three components, each describing a different level of object detection: a) original image (no object detection), b) bounding box, and c) instance segmentation.

We evaluate the performance of three keypoint descriptors, namely: SIFT, RootSIFT [2], and SuperPoint [12] (SP). For each of the descriptors we match the keypoints using brute force (BF). Additionally, with the SuperPoint keypoints we also assess the graph-based matching algorithm SuperGlue [30] (SG) in two constellations: a model trained on indoor data (SGIndoor) and a model trained on outdoor images (SGOutdoor). Next, we clean the matches presented by the brute force method using the distance ratio test proposed by Lowe [19]. We do not further process the matches presented by SuperGlue as it already contains a cleaning step. In total we end up with 15 models based on the aforementioned variations to the pipeline. Lastly, we rank the images based on the number of matches under the general assumption that more keypoint matches indicates a higher visual similarity.

Evaluation on the TinyMola Dataset

In Paper G we evaluate our models on the TinyMola dataset, which contains 91 images and 29 individuals from the Match My Mola database. The evaluation is based on the mean average precision (mAP) and in Figure 4.6 we show the results for all the models. Note, the images of the TinyMola dataset ranges from 0.1 to 16 mega pixel in resolution and varies widely in color due to the underwater environment, therefore, as a pre-processing step we normalize the images by converting them to gray scale and resize them to 640×480.

We find that the handcrafted feature descriptors, SIFT and RootSIFT, basically provide random guesses and are not suitable for re-identifying giant sunfish in the pipeline that we propose. On the other hand, SuperPoint shows promising performance with an mAP of 50-60% at full rank. The matching algorithm has an effect on the performance of SuperPoint and we see that the SuperGlue model trained on indoor images is inferior compared to the others. The brute force approach is the better option for the segmentation

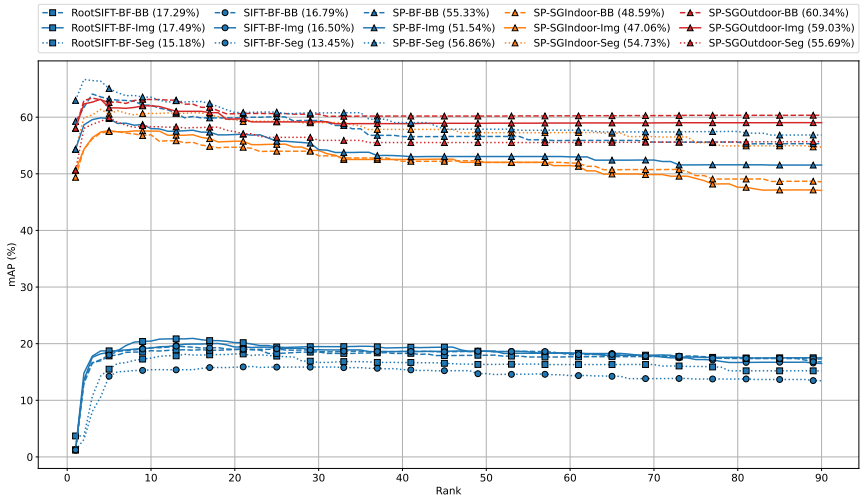


Fig. 4.6: *"**Evaluation results.** The legends specify descriptor-matching-segmentation, e.g., SP-SGIndoor-BB is a combination of the SuperPoint descriptor, SuperGlue indoor matching, and bounding box segmentation. The handcrafted feature descriptors (SIFT and RootSIFT) show weak performance compared to the deep learning based SuperPoint descriptor. The difference between using brute-force matching and the graph-based SuperGlue is less profound and the segmentation level seems to affect the performance ambiguously. The mAP presented in the legends are from the last rank."* [25], figure is from [25], Paper G.

level matches, but the opposite is true for the original image and bounding boxes where SGOutdoor is the best. The segmentation level seem to have an ambiguous effect, which is likely due to mainly two things. The segmentation model is an ImageNet pre-trained Mask R-CNN fine-tuned on 100 manually annotated sunfish images (which are not part of the TinyMola dataset). It achieves a good, but not perfect, performance with an average precision of 88%. This means that some sunfish may be segmented inappropriately which also affects the bounding boxes as they are enclosing the segmentation masks. Additionally, the images are captured in different environments meaning that correct keypoint matches will mainly be located on the sunfish, except for rare cases, e.g., when multiple distinct fish are occurring in the same image. Hence, keypoint descriptors that are not susceptible to noise should be minimally influenced by the surroundings, while an erroneous segmentation may leave out critical information.

To summarize, we conclude that SuperPoint is superior to the hand-crafted feature descriptors in the pipeline proposed in Paper G. We find that the level of object detection has an ambiguous effect on the re-identification performance. Lastly, the SuperGlue matching algorithm seems to increase the performance only when trained on certain types of data compared to using a brute force approach.

From Theory to Practice

At the time when the we proposed the pipeline in Paper G the Match My Mola database contained approximately 1,000 so-called 'photo events', which are recordings of giant sunfish by different divers, locations, or periods. Each of the photo events consisted of 1-3 images of one or both sides of the fish. The marine biologists curating the database were engaged in their work on manually matching all the images of the database and were interested in using our approach for assistance. Therefore, we processed and compared all the images from the Match My Mola database using the pipeline of Paper G configured with the SuperPoint feature descriptor and the SGOutdoor matching model. The biologists received the top-20 ranked images for each of the input images to manually verify potential matches. Despite that several new matches were discovered and the matching time was significantly reduced for a part of the task the approach had practical flaws.

It was a time-consuming and tedious task to manually sort out a very large number of false positives. It is likely that a majority of the sunfish in the database has only been captured on an image once, while others may have been recorded many times, therefore, it is a non-trivial task to choose the optimal number of images to present to the user. We discussed with the curators of the Match My Mola database, and found that they were especially interested in two things for the next iteration of the system, namely: a limited number of ranked images and better visuals when conducting the manual verification, e.g., in the form of colored overlays of the body-patterns, greater contrasts, or the likes.

A Contrastive Re-identification Approach

Based on the key topics proposed by the Match My Mola researchers we investigated ways of including stronger visuals and minimizing the number of false positives. One of the problems with creating an overlay to highlight the unique markings is the huge variability within the appearance of the markings between images and individuals. Whether using traditional image processing techniques like blob-analysis or state-of-the-art semantic segmentation models, it requires training and will likely suffer from limited generalizability. However, enhancing the image contrast would be a straightforward approach to ease manual verification by making the patterns more pronounced. Therefore, as part of the second iteration of the pipeline, presented in Paper H, we included a pre-processing module that enhanced the contrast of the images using the 'contrast limited adaptive histogram equalization' method (CLAHE) [27]. Two images of the same individual captured at different times are presented in Figure 4.7, the left column shows the originals and the right column shows the contrast enhanced images. The squares highlight an easy recognisable pattern.

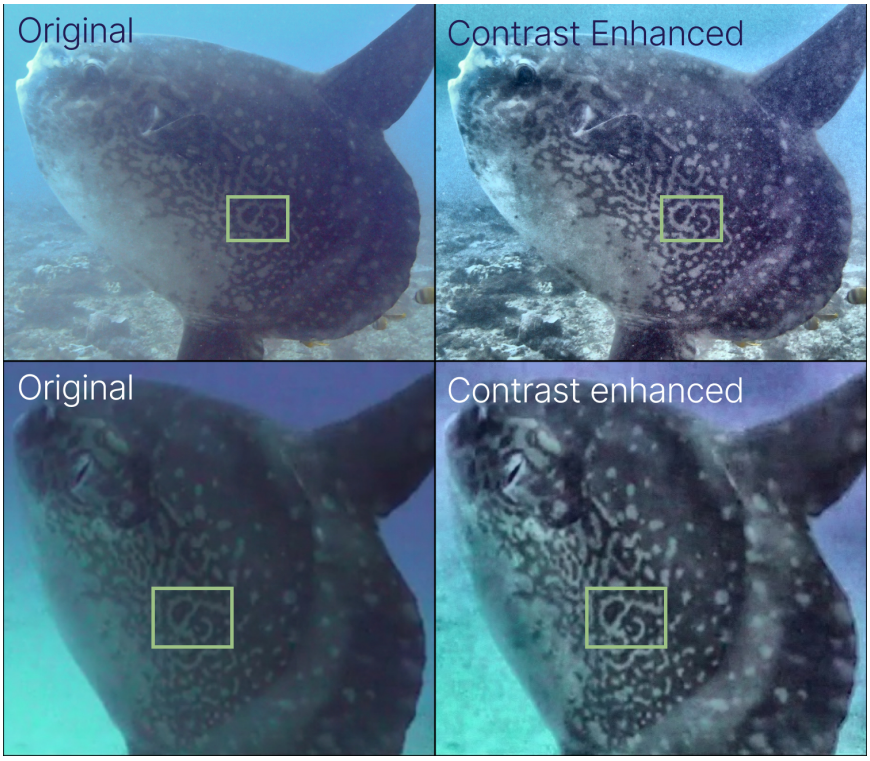


Fig. 4.7: Example of using contrast enhancement to make the markings on the sunfish more bold and thereby ease the manual verification step.

Generally, the contrast enhancement makes it easier to manually match the sunfish in two images, however, this will at most have a limited effect on the time spent on the manual verification procedure. Therefore, inspired by the image registration field, we propose an additional step which is to compute the homography based on the keypoint matches between to images and align the images according to the estimated projective transformation. As previously mentioned, the body of the sunfish is rigid and approximates a 2D plane. This means that if two images contain the same individual we expect a limited projective transform comparable to a natural shift in perspective due to a change in camera position or movement of the object. In the other end of the spectrum, false positive keypoint matches between non-identical individuals most likely do not agree on a simple transformation, and, therefore, lead to a crooked homography estimation. In Figure 4.8 we present a probe image, a set of gallery images, and projections of the probe with respect to the gallery images. The green check mark indicates that the gallery image contains the same individual as the probe while the red cross means another individual.

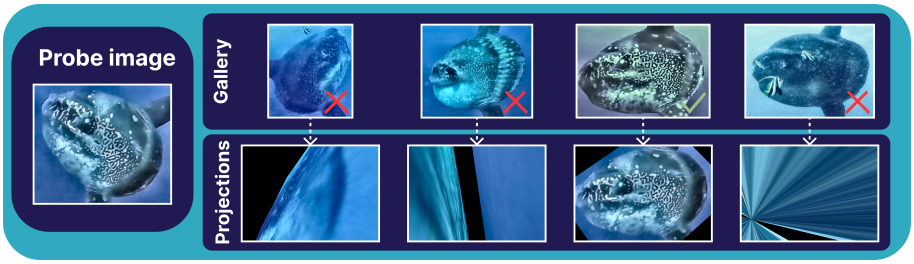


Fig. 4.8: The probe image is projected for each of the gallery images using the homography calculated from the set of SIFT keypoint matches between the probe and gallery image. Note that the image with the green check mark contains the same individual as the probe and it is the only case where the projected probe image is well-aligned with the gallery image.

Estimating the homography and projecting the probe image with respect to the top-n ranked candidates provides an alternative and significantly faster approach to conduct the manual verification step. However, this does not minimize the number of false positives, which was another problem mentioned by the Match My Mola researchers. The problem of minimizing the number of false positives lies in the way most re-identification methods are designed and evaluated, namely based on their ranking capabilities. Therefore, in Paper H, as an alternative, we propose to view the re-identification task as a binary classification problem and state whether a pair of images contain the same individual or not. This can be achieved with a straightforward addition to the proposed pipeline, namely by including a threshold that states that every pair of images with at least x number of keypoint matches is considered a positive match and the rest are considered negatives.

In Paper H we show that the pipeline configured with the SuperPoint feature descriptor and a threshold of minimum 25 keypoint matches reaches a recall of 0.44 and precision of 0.98 on the TinyMola+ dataset, which contains 224 images and 83 IDs. The recall is not staggering, but a precision close to 1.0 basically eliminates the need for human verification of almost half the dataset. For comparison, configuring the pipeline with a pre-processing step using CLAHE, the RootSIFT feature descriptor, and a threshold of minimum 120 keypoint matches we are able to get a recall of 0.34 and a precision of 0.99 illustrating the superiority of SuperPoint.

A combination of the projective transformation approach, for enhancing the visual inspection procedure, with the binary classification approach may prove to be a strong setup for a human-in-the-loop system for conservation efforts where data is limited.

3 Summary and Scientific Contributions

The giant sunfish *Mola alexandrini* is an elusive fish that is rarely seen and we generally know little about it, despite that it is the world's heaviest bony fish. It has unique patterns on its body that allow for re-identification across years. In Paper G we proposed a re-identification pipeline based on keypoint matching and evaluated the pipeline in 15 different configurations based on the level of object detection, choice of feature descriptor, and matching algorithm. The pipeline consisted of public available methods implemented with default settings, without training or fine-tuning any parameters. We were able to achieve a mean average precision (mAP) of around 60% on the TinyMola dataset containing 91 images.

We expanded upon this work in Paper H proposing solutions actively aimed at solving two practical issues expressed by the marine researchers curating the Match My Mola database, namely: easing the visual verification procedure and minimizing the number of false positives. We included a contrast enhancement step to make the patterns more bold, which also had a positive effect on the performance of the pipeline when configured with traditional handcrafted feature descriptors. Additionally, we suggested to estimate the homography between image pairs and subsequently align the images by applying the projective transformation. The projected image will appear crooked and misaligned for image pairs featuring different individuals, offering a clear and rapid approach to eliminate potential false positive matches. Lastly, we discussed the issue of having an inherent high number of false positives when viewing the re-identification task as a ranking problem where it is non-trivial to decide on an optimal number of proposals (top-n rank). In extension thereof, we proposed an alternative output module that incorporated a threshold, transforming the re-identification task into a binary classification problem. This modification eliminated practically all false positives, achieving a precision and recall of 0.98 and 0.44, respectively, on the TinyMola+ dataset containing 224 images.

The main scientific contributions in this chapter can be summarized as:

- In Paper G we proposed a pipeline, requiring no training or parameter fine-tuning, for re-identifying giant sunfish. We illustrated that the deep learning based and state-of-the-art SuperPoint keypoint descriptor is both suitable for this specific task and superior to the traditional handcrafted feature descriptors SIFT and RootSIFT.

- In Paper H we introduced a contrast enhancement module and showed that, beside its practical use, it effectively increases the performance of the SIFT and RootSIFT keypoint descriptors. Furthermore, we proposed a practical solution to ease manual verification by image-alignment based on homography estimation.

- In Paper H we also discussed the in-practicality of having a ranked list as output of a re-identification system and suggested to view the task as a binary classification problem. We proposed an alternative binary output module, effectively eliminating the problem of a high number of false positives induced by the non-trivial task of choosing an optimal number of ranked proposals.

Future Work

The methods proposed in Paper G and Paper H have been developed with practicality in mind, as solutions to a real problem. Therefore, it is imperative for future development to maintain the close collaboration with the users of the system, the marine biologists and curators of the Match My Mola database. An obvious and relevant piece of future work would, therefore, be to investigate the pros and cons of using the pipeline as part of a human-in-the-loop system for processing new images submitted to the Match My Mola database. Interesting questions include: 1) What is the level of accuracy? 2) How much time is spent on the task? 3) How does the duration of consecutive time spent on the task affect the performance? One approach to carry out such experiments would involve dividing a number of participants into two groups: one group following the traditional protocol for visual inspection, and another group utilizing our proposed system. By comparing the outcomes and the time invested, we can assess the effectiveness and efficiency of each approach.

When it comes to improving the existing system or proposing a new one, an approach to consider is the integration of supplementary data from the Match My Mola database, including information about the location and time of recording. This may be applicable to exclude potential false positives using basic heuristics. For example, it can be assumed that an individual sunfish cannot travel extreme distances within a short period of time.

In a broader sense, exploring approaches to utilize the existing binary pipeline as an initial stage for obtaining pseudo-labelled data to train supervised deep neural networks would be interesting. This could for example be a neural network trained with triplet loss, which has been shown to be suitable for animal re-id of patterned species [32]. It is most likely that a neural network learns to distinguish the fish based on a different set of features compared to the keypoint descriptor used to collect the pseudo-labelled data. Therefore, it is also plausible that the network is capable of finding other matches and improve the performance of the system.

References

[1] W. Andrew, S. Hannuna, N. Campbell, and T. Burghardt, "Automatic individual holstein friesian cattle identification via selective local coat pattern matching in RGB-d imagery," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, sep 2016.

[2] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 2911–2918.

[3] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Computer Vision – ECCV 2006*. Springer, 2006, pp. 404–417.

[4] C. Bergler, A. Gebhard, J. R. Towers, L. Butyrev, G. J. Sutton, T. J. H. Shaw, A. Maier, and E. Nöth, "FIN-PRINT a fully-automated multi-stage deep-learning-based framework for the individual recognition of killer whales," *Scientific Reports*, vol. 11, no. 1, dec 2021.

[5] D. T. Bolger, T. A. Morrison, B. Vance, D. Lee, and H. Farid, "A computer-assisted system for photographic mark-recapture analysis," *Methods in Ecology and Evolution*, vol. 3, no. 5, pp. 813–822, may 2012.

[6] S. Bouma, M. D. Pawley, K. Hupman, and A. Gilman, "Individual common dolphin identification via metric embedding learning," in *International conference on image and vision computing New Zealand (IVCNZ)*. IEEE, 2018, pp. 1–6.

[7] A. Castro Cabanillas and V. H. Ayma, "Humpback whale's flukes segmentation algorithms," in *Annual International Conference on Information Management and Big Data*. Springer, 2020, pp. 291–303.

[8] I. Chelak, E. Nepovinnykh, T. Eerola, H. Kälviäinen, and I. Belykh, "EDEN: Deep feature distribution pooling for saimaa ringed seals pattern matching," in *Cyber-Physical Systems and Control II*. Springer International Publishing, 2023, pp. 141–150.

[9] J. P. Crall, C. V. Stewart, T. Y. Berger-Wolf, D. I. Rubenstein, and S. R. Sundaresan, "Hotspotter—patterned species instance recognition," in *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE, 2013, pp. 230–237.

[10] J. Davenport, N. D. Phillips, E. Cotter, L. E. Eagling, and J. D. R. Houghton, "The locomotor system of the ocean sunfish *Mola mola* (l.): role of gelatinous exoskeleton, horizontal septum, muscles and tendons," *Journal of Anatomy*, vol. 233, no. 3, pp. 347–357, Jun. 2018.

[11] P. De Zeeuw, E. Pauwels, E. Ranguelova, D. Buonantony, and S. Eckert, "Computer assisted photo identification of dermochelys coriacea," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*. Springer, 2010, pp. 165–172.

[12] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 224–236.

[13] S. G. Dunbar, E. C. Anger, J. R. Parham, C. Kingen, M. K. Wright, C. T. Hayes, S. Safi, J. Holmberg, L. Salinas, and D. S. Baumbach, "HotSpotter: Using a computer-driven photo-id application to identify sea turtles," *Journal of Experimental Marine Biology and Ecology*, vol. 535, p. 151490, feb 2021.

[14] J. N. Gomes-Pereira, C. K. Pham, J. Miodonski, M. A. R. Santos, G. Dionísio, D. Catarino, M. Nyegaard, E. Sawai, G. P. Carreira, and P. Afonso, "The heaviest bony fish in the world: A 2744-kg giant sunfish mola alexandrini (ranzani, 1839) from the north atlantic," *Journal of Fish Biology*, vol. 102, no. 1, pp. 290–293, nov 2022.

[15] C. Gope, N. Kehtarnavaz, G. Hillman, and B. Würsig, "An affine invariant curve matching method for photo-identification of marine mammals," *Pattern Recognition*, vol. 38, no. 1, pp. 125–132, jan 2005.

[16] C. L. Huffard, R. L. Caldwell, N. DeLoach, D. W. Gentry, P. Humann, B. MacDonald, B. Moore, R. Ross, T. Uno, and S. Wong, "Individually unique body color patterns in octopus (wunderpus photogenicus) allow for photoidentification," *PLoS ONE*, vol. 3, no. 11, p. e3732, Nov. 2008.

[17] S. Li, J. Li, H. Tang, R. Qian, and W. Lin, "ATRW: A benchmark for amur tiger re-identification in the wild," in *Proceedings of the 28th ACM International Conference on Multimedia*. ACM, oct 2020.

[18] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, vol. 2. Ieee, 1999, pp. 1150–1157.

[19] ——, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[20] R. Maglietta, V. Renò, G. Cipriano, C. Fanizza, A. Milella, E. Stella, and R. Carlucci, "DolFin: an innovative digital platform for studying risso's dolphins in the northern ionian sea (north-eastern central mediterranean)," *Scientific Reports*, vol. 8, no. 1, nov 2018.

[21] E. Nepovinnykh, T. Eerola, and H. Kalviainen, "Siamese network based pelage pattern matching for ringed seal re-identification," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, 2020, pp. 25–34.

[22] M. Nyegaard, "There be giants! the importance of taxonomic clarity of the large ocean sunfishes (genus mola, family molidae) for assessing sunfish vulnerability to anthropogenic pressures." Ph.D. dissertation, Murdoch University, 2018. [Online]. Available: http://researchrepository.murdoch.edu.au/id/eprint/41666

[23] M. Nyegaard, J. Karmy, L. McBride, T. M. Thys, M. Welly, and R. Djohani, "Rapid physiological colouration change is a challenge - but not a hindrance - to successful photo identification of giant sunfish (mola alexandrini, molidae)," *Frontiers in Marine Science*, vol. 10, may 2023.

[24] Ocean Sunfish Research Trust, "Match my mola," https://oceansunfishresearch.org/matchmymola/, accessed: 2021-09-21.

[25] M. Pedersen, J. B. Haurum, T. B. Moeslund, and M. Nyegaard, "Re-identification of giant sunfish using keypoint matching," vol. 3, 3 2022.

[26] M. Pedersen, M. Nyegaard, and T. B. Moeslund, "Finding nemo's giant cousin: Keypoint matching for robust re-identification of giant sunfish," *Journal of Marine Science and Engineering*, vol. 11, no. 5, 2023.

[27] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive histogram equalization and its variations," *Computer Vision, Graphics, and Image Processing*, vol. 39, no. 3, pp. 355–368, September 1987.

[28] A. J. Richardson, A. Bakun, G. C. Hays, and M. J. Gibbons, "The jellyfish joyride: causes, consequences and management responses to a more gelatinous future," *Trends in Ecology & Evolution*, vol. 24, no. 6, pp. 312–322, 2009.

[29] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Ieee. IEEE, November 2011, pp. 2564–2571.

[30] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 4938–4947.

[31] E. Sawai, Y. Yamanoue, M. Nyegaard, and Y. Sakai, "Redescription of the bumphead sunfish mola alexandrini (ranzani 1839), senior synonym of mola ramsayi (giglioli 1883), with designation of a neotype for mola mola (linnaeus 1758)(tetraodontiformes: Molidae)," *Ichthyological Research*, vol. 65, no. 1, pp. 142–160, 2018.

[32] S. Schneider, G. W. Taylor, and S. C. Kremer, "Similarity learning networks for animal individual re-identification - beyond the capabilities of a human observer," in *Proceedings of the IEEE Winter Applications of Computer Vision Workshops (WACVW)*, 2020, pp. 44–52.

[33] S. Schneider, G. W. Taylor, S. Linquist, and S. C. Kremer, "Past, present and future approaches using computer vision for animal re-identification from camera trap data," *Methods in Ecology and Evolution*, vol. 10, no. 4, pp. 461–470, jan 2019.

[34] C. W. Speed, M. G. Meekan, and C. J. Bradshaw, "Spot the match–wildlife photo-identification using information theory," *Frontiers in zoology*, vol. 4, no. 1, pp. 1–11, 2007.

[35] M. C. Stoddard, R. M. Kilner, and C. Town, "Pattern recognition algorithm reveals how birds evolve individual egg pattern signatures," *Nature Communications*, vol. 5, no. 1, jun 2014.

[36] C. Town, A. Marshall, and N. Sethasathien, "Manta matcher: automated photographic identification of manta rays using keypoint features," *Ecology and Evolution*, vol. 3, no. 7, pp. 1902–1914, may 2013.

[37] D. Tuia, B. Kellenberger, S. Beery, B. R. Costelloe, S. Zuffi, B. Risse, A. Mathis, M. W. Mathis, F. van Langevelde, T. Burghardt, R. Kays, H. Klinck, M. Wikelski, I. D. Couzin, G. van Horn, M. C. Crofoot, C. V. Stewart, and T. Berger-Wolf, "Perspectives in machine learning for wildlife conservation," *Nature Communications*, vol. 13, no. 1, feb 2022.

[38] W. Wang, R. Solovyev, A. Stempkovsky, D. Telpukhov, and A. Volkov, "Method for whale re-identification based on siamese nets and adversarial training," *Optical Memory and Neural Networks*, vol. 29, no. 2, pp. 118–132, 2020.

[39] L. Zhao, M. Pedersen, J. Y. Hardeberg, and B. Dervo, "Image-based recognition of individual trouts in the wild," in *Proceedings of the European Workshop on Visual Information Processing (EUVIP)*. IEEE, 2019.

Chapter 5

Conclusion

This thesis revolves around the task of automating monitoring of marine organisms. The main objective has been to investigate and advance the field of computer vision in underwater settings, specifically focusing on the research question: What are the limitations and challenges associated with detecting and tracking fish in diverse underwater environments using computer vision, and how can they be overcome?

In the past decade, there has been an increasing emphasis on conservation efforts in the marine sector. To effectively target these efforts and maximize their impact, marine scientists have been encouraged to gather data at an unprecedented pace. However, traditional marine monitoring approaches are often harmful or intrusive due to the necessity of catching or tagging the organisms. Additionally, they are typically not scalable as they require a large deal of manual intervention. Underwater cameras and computer vision techniques provide scalable, non-intrusive, and objective solutions to gather and analyse data efficiently. However, the large variation and complexity of underwater environments put great demands on the algorithms that need to be versatile and robust to effectively cope with the dynamic and diverse nature of marine habitats. Deep learning algorithms are anticipated to possess the capacity to tackle these challenges due to their inherent design, abstraction capabilities, and demonstrated success in various other domains.

During the past decade deep learning algorithms have matured to a degree where they are now actively being used for conservation efforts. Currently, this is mainly in the terrestrial domain, likely due to the difficulties of acquiring high quality visual data from underwater environments. The latter has entailed a critical lack of visual marine benchmark datasets, and it has basically not been possible to evaluate algorithms aimed at the underwater domain on a credible foundation. Therefore, a significant part of this work has been focused at addressing the lack of publicly available annotated underwater datasets.

We have chosen a precautionary approach to the problem by conducting initial experiments in a controlled environment before venturing into the wild. A major contribution of this work is the development of three underwater benchmark datasets, namely 3D-ZeF, the Brackish dataset, and BrackishMOT. 3D-ZeF was the first publicly available underwater 3D multi-object tracking dataset. The dataset contains sequences with multiple zebrafish in an aquarium recorded from a stereo setup, which comprises that the fish needs to be detected and associated from two perspectives: one camera is placed above the aquarium and the second camera is placed in front of the aquarium. 3D-ZeF presents a great challenge for the broader tracking community as the task is to detect similarly looking, social, and erratically moving objects in two views, associate the observations, and reconstruct the 3D trajectories of the objects. Based on the experience gained from developing 3D-ZeF, and to pave the way for creating transparent and balanced multi-object tracking datasets, we developed an objective complexity metric named MOTCOM to estimate the difficulty of multi-object tracking sequences with respect to occlusion, visual similarity, and erratic motion. Moving from a controlled environment and into the wild; we developed the Brackish dataset recorded from a stationary camera system mounted in the local strait 'Limfjorden' in the northern part of Jutland, Denmark. It was the first object detection dataset of marine organisms acquired in brackish water and with varying visibility. It contains sequences that are significantly different from other underwater datasets, which have typically been captured in tropical or clear waters. The Brackish dataset laid the foundation for the multi-object tracking expansion named BrackishMOT, which contained additional sequences and multi-object tracking annotations. BrackishMOT was among the first underwater multi-object tracking datasets captured in the wild and presents a significant challenge due to the focus on groups of social and similarly looking fish. MOTCOM was actively used in the design of the BrackishMOT dataset to ensure that the training and test splits where balanced with respect to tracking complexity.

Beside publishing datasets, we have also developed and evaluated novel solutions and assessed state-of-the-art methods on the data. Along with the 3D-ZeF dataset we also proposed a 3D multi-object tracking pipeline following a tracking-reconstruction approach, where the objects are detected and associated into tracks for each view independently, before reconstructing the 3D positions based on a between-view association of the 2D tracks. We intentionally designed the pipeline in a modular fashion to facilitate the replacement of components, such as the 2D object detection step, which has historically been a rapidly progressing field in computer vision. Regarding the Brackish and BrackishMOT datasets, we investigated ways of fine-tuning and adapting state-of-the-art models to the brackish and turbid underwater environment. We found that it is feasible to fine-tune deep learning models for detecting and tracking marine organisms, but due to the stationary camera care must be

taken to avoid overfitting the models. Additionally, we proposed a framework for generating realistic-looking synthetic data and showed that it is achievable to teach a model to track fish from synthetic data, which has great potential for creating more robust and general models without acquiring and annotating large amounts of underwater data. Lastly, we proposed a solution for re-identification of giant sunfish, which is a large fish with unique patterns, that allows for identifying the same individual across images. Currently, visual identification is conducted manually, which is difficult and extremely time-consuming. There are only few verified matches, therefore, this task is not suited for a supervised model. Instead, we presented a pipeline based on key-point matching and consisting purely of conventional and publicly available algorithms using default settings. Our evaluation showed that the proposed solution achieved promising results on a private dataset with giant sunfish without any training or parameter fine-tuning. Moreover, we proposed to use image-alignment based on homography estimation to ease the manual verification process, and discussed the in-practicality of having a ranked output due to the large number of inherent false positives as a result of the non-trivial problem of choosing an optimal number of proposals.

Future Work

Key elements that highlight potential directions for future work is presented here. For more in-depth explanations and discussions, please see the future work sections concluding each of the three chapters.

Chapter 2: Tracking Multiple Fish in a Controlled Environment

- The 3D tracker developed for the 3D-ZeF dataset contains a 2D detection module, where immediate gain can be achieved by implementing a stronger object detector. Additionally, the 3D reconstruction module of the system provides information on the uncertainty of the estimated 3D positions in the form of a reprojection error, which may be used actively to refine detections or even estimate the position of objects that are occluded or undetected in one of the two views.
- We proved that MOTCOM is superior at estimating the complexity of multi-object tracking sequences compared to conventional metrics based on the number of objects. However, the evaluation was based on a single domain of MOT, namely, pedestrian tracking. It would be highly relevant to include data from other tracking domains to cement the generalizability of the algorithm. Additionally, it would be interesting to investigate how an imbalance in the complexity of train and test splits affects the estimated performance of trackers.

Chapter 3: Monitoring Marine Organisms in the Wild

- The Brackish dataset has been captured in 'Limfjorden' from a single stationary camera, which has the disadvantage of containing little variation to the surrounding environment. It would increase the practicality of the dataset if we included a mix of stationary and moving recordings from different areas of 'Limfjorden'.
- Due to the difficulty of correctly annotating similarly looking species in the wild it would be relevant to capture images and sequences of fish in realistic-looking aquarium environments, e.g., Oceanarium in Hirtshals or the Blue Planet in Kastrup, to be able to assess fine-grained classification algorithms under semi-wild conditions.

Chapter 4: Long-Term Monitoring of Giant Sunfish

- The pipeline developed for re-identifying giant sunfish did not require training or fine-tuning, nonetheless, we were able to identify almost half of the individuals with close to zero false positives. Therefore, the pipeline may be suitable for acquiring pseudo-labelled data that can be used to train more complex re-identification models.

Final Remarks

Our findings point to a lack of publicly available and high quality underwater benchmark datasets as being a critical limitation to further advancement of automated marine monitoring using computer vision. We consider this to be partially attributed to the demanding marine conditions that impose significant requirements and challenges to preparation, maintenance, and image acquisition in comparison to terrestrial environments. However, we find that present computer vision algorithms are suited for a wide range of underwater applications; from behavioral analysis of fish in a controlled environment to multi-object tracking in brackish water and long-term tracking of individual giant sunfish. Additionally, in recent years there has been a growing focus on the field, which appears to be an ongoing trend. The United Nations has targeted one of their sustainable development goals directly at the sustainability and conservation of our oceans and there is a general drive towards more biodiversity and enhanced governance of Earth's resources. Therefore, we anticipate that computer vision will play an increasingly prominent role in the field of marine monitoring, including targeting future conservation efforts.

Part II

Papers

Paper A

Camera Calibration for Underwater 3D Reconstruction Based on Ray Tracing using Snell's Law

Malte Pedersen, Stefan Hein Bengtson, Rikke Gade,
Niels Madsen, and Thomas B. Moeslund

The paper has been published in the
Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1410-1417, 2018.

The layout has been revised.

Abstract

Accurately estimating the 3D position of underwater objects is of great interest when doing research on marine animals. An inherent problem of 3D reconstruction of underwater positions is the presence of refraction which invalidates the assumption of a single viewpoint. Three ways of performing 3D reconstruction on underwater objects are compared in this work: an approach relying solely on in-air camera calibration, an approach with the camera calibration performed under water and an approach based on ray tracing with Snell's law. As expected, the in-air camera calibration showed to be the most inaccurate as it does not take refraction into account. The precision of the estimated 3D positions based on the underwater camera calibration and the ray tracing based approach were, on the other hand, almost identical. However, the ray tracing based approach is found to be advantageous as it is far more flexible in terms of the calibration procedure due to the decoupling of the intrinsic and extrinsic camera parameters.

1 Introduction

Research on marine animals is becoming increasingly popular in terms of studying pharmacology, genetics and marine ecosystems. Environmental studies being especially popular due to the increased focus on how emission of various pollutants, such as microplastics [1, 2], may affect the environment and us.

One approach to study the impact of marine pollution is to expose a model organism to a pollutant in a controlled environment and analyze the behavioral patterns before and after the exposure.

Mapping these behavioral patterns is a time-consuming process and has therefore inspired multiple vision based systems, which automate this process to some degree [3, 4]. However, a large part of these systems only supports tracking of animals in a single plane; i.e., shallow water when working with marine animals. This is problematic as most marine animals naturally move in three dimensions and studying them in a single plane is insufficient in most cases [5].

Gathering 3D information about marine animals is not without difficulties in vision based systems as the light captured by the cameras will be exposed to refraction as it passes through different media. This is especially true when observing a fish tank with cameras placed outside the tank, which is a common setup as it imposes less restrictions on the placement of the cameras and their resistance to water.

The refraction occurring at the media interfaces, such as the aquarium boundaries, can be described using Snell's law [6] illustrated in Figure A.1. It

relates the angles of incidence and refraction of the light ray by

$$\frac{\sin\theta_1}{\sin\theta_2} = \frac{v_1}{v_2} = \frac{n_2}{n_1}, \tag{A.1}$$

where θ is the angle between the surface normal and light ray, v is the velocity of light, and n is the refractive index of the respective medium.

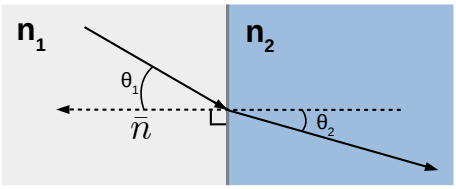


Fig. A.1: Illustration of the relationship between the angles of incidence and refraction described by Snell's law.

How refraction can affect a camera is illustrated in Figure A.2, where multiple light rays travel from water to air until they are captured by the camera lens. The refraction essentially causes the camera to experience the scene as if it was observed from n viewpoints, vp_n, and thereby invalidates the assumption of a single viewpoint which is prevalent in most common camera models, such as the one described in [7].

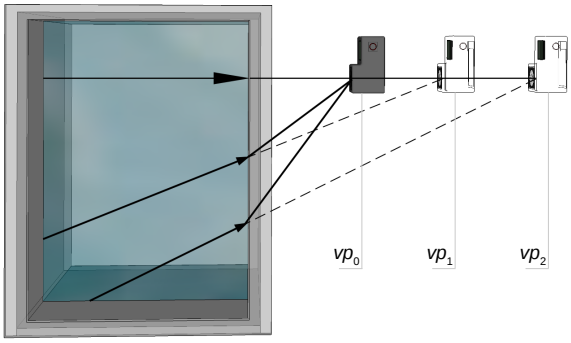


Fig. A.2: Illustration of how light rays are bent due to refraction before entering the aperture of the camera. The transparent cameras represent virtual viewpoints caused by the refraction.

The focus of this paper is to outline and compare the precision of various ways of dealing with refraction when using a stereo vision based system to gather 3D information about marine animals in aquariums.

1.1 Related Work

The different ways of handling refraction when gathering 3D information can generally be divided into two categories: approaches that indirectly account for refraction by relying on the camera model to absorb the errors and approaches that directly try to account for the physics of refraction. These will be discussed in more details below.

Some approaches rely solely on the SVP (single viewpoint) camera model, which is advantageous due to its ease of use, as it is well documented and widely supported in many toolboxes. Its popularity is not without reason, as it is also simple to deal with mathematically, mainly because the camera is described using a linear projective transform.

Examples of using the SVP model for underwater purposes can be found in [8–11]. The latter also tries to account for the problem of refraction by mounting a dome port instead of a flat port in front of the camera. Doing so essentially focuses the refracted rays into a single viewpoint. The drawback of such a solution is the precision required to manufacture the dome and align it properly with the given camera.

The main problem of these approaches is that the base assumption of having a single viewpoint is violated due to the refraction, as shown in Figure A.2. However, it can be argued that the error caused by refraction can be absorbed, to some extend, by the focal length adjustment and radial distortion correction commonly found in SVP models.

This idea is the scope of [12], which explores the performance of the SVP camera model for 3D underwater reconstruction. The paper describes tests of different configurations of the SVP model with and without focal length adjustment and radial distortion correction. It is found that the adjustment of the focal length has the biggest impact on reducing the error caused by refraction.

A variation of this idea is to use multiple localized focal length adjustments instead of a single global focal length adjustment. An example of such an approach can be found in [13] which utilizes a pixel-wise varifocal camera model. Another example is the method presented in [14] which segments the scene into smaller partitions and calibrates localized SVP camera models for each partition.

The second group of approaches are the ones which actively seek to counteract the refraction error by modelling the underlying physics.

A straightforward example of such an approach can be found in [15], where two cameras are used to track a single fish in 3D. The cameras are modeled using the SVP camera model and their intrinsic parameters have been found through calibration in air. The extrinsic parameters of the camera are found in relation to the corners of the aquarium.

The intrinsic and extrinsic parameters are then used to project two rays, one for each camera, into the aquarium. Knowledge of the aquarium's corners are used to calculate the intersection between the rays and the aquarium. The refracted rays originating from the intersections are calculated and used for triangulation.

A somewhat similar approach is described in [16], where multiple cameras are used to track a single seahorse in 3D. This approach relies on the same combination of ray tracing, Snell's law, and the SVP camera model. However, it differs in the sense, that it includes an additional step where the intrinsic and extrinsic camera parameters are optimized. The estimated position and orientation of the refractive interface is also optimized during this step, based on reference points on a known calibration frame. The approach described in [17] performs a similar optimization step to refine the parameters used during ray tracing as well.

Other approaches discard the SVP camera model in favor of an axial camera model [18], as this model only assumes that the light ray will intersect along a common line and not in a common focal point. The use of axial camera models for marine research is however sparse, as they are deemed unpractical [10].

Two of the most frequently used approaches do hence appear to be the SVP camera model using a calibration frame placed under water [8–10] and ray tracing in combination with Snell's law [15–17].

Motivation in the respective works, as to why one approach is used instead of the other, is however lacking, as no clear comparison has been made between the two. The SVP camera model has been popular for many years due to its simplicity and precision and it is therefore well documented and easy accessible. This is not the case with the ray tracing based approach which has not gained a lot of attention; possibly because the demand for precise 3D estimation of objects, in cases where light moves through different media, is not high. Due to this, the contributions of this paper will be

- A comparison between the approach relying on the SVP camera model with an underwater calibration frame and the approach based on ray tracing in combination with Snell's law.

  Furthermore, the impact of refraction is tested by comparing both of these approaches against the SVP camera model using a calibration frame placed in air.

- A description of each step needed to perform ray tracing in combination with Snell's law to account for refraction.

- A publicly available Python implementation of the ray tracing approach.

The rest of the paper is organized with first a thorough explanation of the ray tracing based approach, followed by a description of the evaluation process and lastly a comparison between the mentioned approaches.

2 Ray Tracing using Snell's Law

In this section, the ray tracing based method, to precisely estimate the 3D position of an underwater object placed in a tank of water, will be presented. All the steps are visualized in a simplified manner in Figure A.4, where each number fits the number of the subsection describing the respective step.

The first part describes how to find the intrinsic and extrinsic camera parameters in a way that keeps them separated, which opens for a more flexible setup. This is followed by an outline of how ray tracing can be used in combination with Snell's law to expand the SVP model to account for refraction. By taking refraction into account using ray tracing, the system becomes unaffected by the size of the aquarium as no calibration frame is needed under water. In the end it is explained how the resulting rays can be used to estimate the 3D position of an underwater object using triangulation.

2.1 Camera Calibration

Calibration of the cameras is an essential part of 3D reconstruction as it relates 3D world coordinates to 2D image coordinates. This 3D to 2D relationship can be described by the extrinsic and intrinsic parameters as illustrated in Figure A.3.

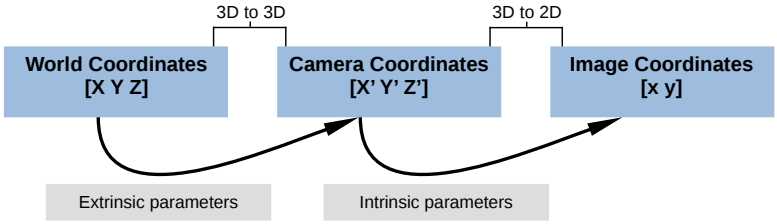


Fig. A.3: Illustration of the relationship between the extrinsic and intrinsic camera parameters.

The extrinsic parameters describe the transformation from world to camera coordinates and the intrinsic parameters describe the relationship between the camera coordinates and the 2D pixel coordinates in the image.

The intrinsic parameters are found for each camera using the method presented in [7], where a checkerboard in air is used for the calibration and 25 images are captured for each camera.

The extrinsic parameters are found using four 3D-2D point correspondences for each camera. The parameters are found by minimizing the reprojection error of the four points in the respective cameras, using an iterative approach based on Levenberg-Marquardt optimization [19]. This approach relies on the intrinsic parameters, as they are used in the calculation of the

reprojection error, namely the 3D to 2D transformation between camera coordinates and image coordinates. The needed point correspondences are found through manual annotation of the intersections between the water surface and the corners of the aquarium.

2.2 Projecting a 2D Point Into a Ray

The second step is to project the 2D image coordinates into rays in the world space coordinate system. This step is repeated for each camera with each of the rays being characterized as

$$r(\lambda) = \lambda \bar{r} + r_0, \tag{A.2}$$

where $\bar{r}$ is the direction of the ray, r_0 is a point on the ray and λ defines all positions along the ray. The direction vector, $\bar{r}$, is found by

$$\bar{r} = R^{-1}K^{-1}\begin{bmatrix} x & y & 1 \end{bmatrix}^T, \tag{A.3}$$

where R^{-1} is the inverse rotation matrix of the camera, K^{-1} is the inverse intrinsic camera matrix and $\begin{bmatrix} x & y \end{bmatrix}^T$ are the image coordinates of the 2D point to back project into a ray.

The point along the ray, r_0, is set to the camera center as the back projected ray must pass through this position, which can be found by

$$r_0 = -R^{-1}t, \tag{A.4}$$

where t is the translation vector of the extrinsic parameters.

2.3 Identifying the Plane-Ray Intersection

The point of intersection between the ray and the plane of the media interface, e.g., the plane separating air and water, must be found in order to account for refraction. Determining this point is essentially a matter of identifying λ_0, such that $r(\lambda_0) = p$, where p is a point on the plane. This point, p, must satisfy the plane equation:

$$(p - p_0) \cdot \bar{n} = 0, \tag{A.5}$$

where $\bar{n}$ is the plane normal and p_0 is an already known point on the plane. Combining Equation (A.2) and Equation (A.5) then yields

$$\lambda_0 = \frac{(p_0 - r_0) \cdot \bar{n}}{\bar{r} \cdot \bar{n}}. \tag{A.6}$$

The intersection point between the plane and ray, I, can then be found by inserting λ_0 into Equation (A.2)

$$I = \lambda_0 \bar{r} + r_0. \tag{A.7}$$

The above calculations require knowledge of the plane normal, $\bar{n}$, which can be found as $\bar{n} = \bar{v_1} \times \bar{v_2}$. The vectors $\bar{v_1}$ and $\bar{v_2}$ being two vectors on the plane, which can be found from three non-collinear points on the plane. These three points are extracted from the set of points manually annotated during the camera calibration. The required known point on the plane, p_0, can be selected from these manually annotated points as well.

2.4 Calculating the Refracted Rays

The refraction of the ray, $r(\lambda)$, at the intersection between the media is calculated using Snell's law. The following describes the steps to calculate the refracted vector, $\bar{r}'$, of an incoming vector, $\bar{r}$, and is based on [20]:

1. Calculate the cosine of θ_1 as

$$cos(\theta_1) = -\bar{n} \cdot \bar{r}, \tag{A.8}$$

   where θ_1 is the angle between $\bar{r}$ and the surface normal of the interface between the media, $\bar{n}$.

2. Calculate the cosine of θ_2 as

$$cos(\theta_2) = \sqrt{1 - \left(\frac{n_1}{n_2}\right)^2 (1 - cos(\theta_1)^2)}, \tag{A.9}$$

   where θ_2 is the angle between $\bar{r}'$ and $\bar{n}$.

3. The refracted vector, $\bar{r}'$, can then be described as

$$\bar{r}' = \left(\frac{n_1}{n_2}\right) \bar{r} + \left(\frac{n_1}{n_2} cos(\theta_1) - cos(\theta_2)\right) \bar{n} \tag{A.10}$$

   where n_1 is the refractive index of the medium that the traced ray passes from and n_2 is the refractive index of the medium that the ray passes to.

For a camera placed in front of an aquarium, the indices $n_1 = 1.0$ and $n_2 = 1.33$ are used for air and water, respectively. The refraction caused by the plastic or glass sides of the aquarium is not taken into account, as the impact is minor except for large tanks where the sides are very thick.

2.5 Triangulation using Rays

The final step is to triangulate the 3D position of 2D image coordinates using the refracted rays, $r_1(\lambda)'$ and $r_2(\lambda)'$, from the respective cameras. The refracted rays still adheres to Equation (A.2) and are formed using the refracted direction vector, $\bar{r}'$, from Equation (A.10) along with the position of the plane-ray intersection, I, from Equation (A.7).

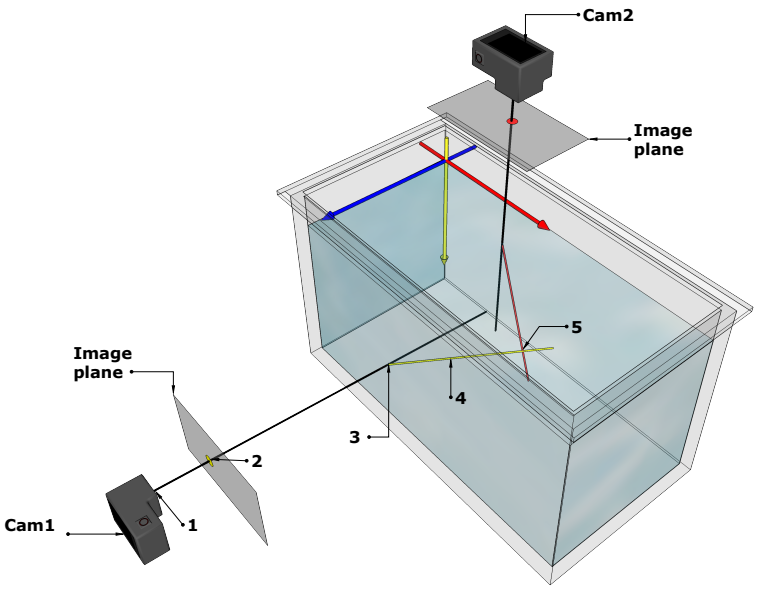


Fig. A.4: Illustration of the five 3D reconstruction steps. **1.** Camera calibration **2.** Projecting a 2D point into a ray **3.** Identifying the plane-ray intersection **4.** Calculating the refracted rays **5.** Triangulation using rays.

The triangulation method employed is commonly known as the midpoint algorithm. The idea is to identify the vector, $\bar{m}$, between the two rays, $r_1(\lambda)'$ and $r_2(\lambda)'$, such that the length, $||\bar{m}||$, is minimized. The final 3D position is found as the midpoint of the vector $\bar{m}$.

Other ways of triangulating 3D positions exist, such as the methods mentioned in [21]. However, the midpoint algorithm is chosen over other triangulation methods, as it operates on rays by default. Other approaches utilizes the camera matrix (formed by the intrinsic and extrinsic camera parameters) to triangulate a point, while minimizing the reprojection error. However, this assumes that the pinhole camera model holds, i.e., a single viewpoint exists, which is not the case as discussed earlier.

The midpoint algorithm is based on the fact that the length, $||\bar{m}||$, must be at its minimum when $\bar{m}$ is perpendicular to both rays. The main idea is hence to identify a vector, $\bar{m}$, such that

$$\begin{aligned} \bar{m} \cdot \bar{r}_1' &= 0 \\ \bar{m} \cdot \bar{r}_2' &= 0, \end{aligned} \tag{A.11}$$

where $\bar{r}_1'$ and $\bar{r}_2'$ are the direction vectors of the two refracted rays and $\cdot$ is the dot product.

The vector, $\bar{m}$, is found by calculating the vector's start position, M_1, and end position, M_2, along the refracted rays. The two points can be calculated by

$$M_1 = I_1 + \bar{r}_1' \frac{-(\bar{r}_1' \cdot \bar{r}_2')(\bar{r}_2' \cdot I_1\bar{I}_2) + (\bar{r}_1' \cdot I_1\bar{I}_2)(\bar{r}_2' \cdot \bar{r}_2')}{(\bar{r}_1' \cdot \bar{r}_1')(\bar{r}_2' \cdot \bar{r}_2') - (\bar{r}_1' \cdot \bar{r}_2')(\bar{r}_1' \cdot \bar{r}_2')} \tag{A.12}$$

and

$$M_2 = I_2 + \bar{r}_2' \frac{(\bar{r}_1' \cdot \bar{r}_2')(\bar{r}_1' \cdot I_1\bar{I}_2) - (\bar{r}_2' \cdot I_1\bar{I}_2)(\bar{r}_1' \cdot \bar{r}_1')}{(\bar{r}_1' \cdot \bar{r}_1')(\bar{r}_2' \cdot \bar{r}_2') - (\bar{r}_1' \cdot \bar{r}_2')(\bar{r}_1' \cdot \bar{r}_2')}. \tag{A.13}$$

where I_1 and I_2 are the plane-ray intersections of the two views.

The final 3D position, P, of the triangulation process, is then calculated as

$$P = \frac{(M_1 + M_2)}{2}. \tag{A.14}$$

3 Evaluation

The described ray tracing approach is compared against two other approaches: calibration using a checkerboard placed in air and calibration using a checkerboard placed under water in the aquarium. Both methods are based on the calibration approach described in [7].

The test is performed by moving a calibration rod, with two brightly colored balls mounted on it, around in an aquarium with water. The rod has a bend near the location of the balls in order to make it easier to move it around without interfering with the cameras, i.e., blocking their field of view or moving them. The cameras are placed as shown in Figure A.5, which is a setup that has been used in several marine life behavioral analysis systems based on stereo vision [9, 15, 16].

The length of the aquarium is 40 cm and it has a width and height of 20 cm and 25 cm, respectively. A total of 15.6 liters of water, equalling a depth of 19.5 cm, has been used for the test.

The colored balls are used as markers and all three approaches have been evaluated by their euclidean inter-distance error, e_i, given by

$$e_i = d_w - |p_1' - p_2'|, \tag{A.15}$$

where d_w is the known distance between the two markers, while $|p_1' - p_2'|$ is the euclidean distance between the markers when their 3D positions have been estimated. This error will hence provide a measure of the ability of each approach to correctly judge distances while capturing underwater objects.

The actual triangulation steps remain unchanged throughout the test and are as described in Section 2.5. The main difference between the tested approaches is hence how the camera calibration is performed and how the rays for the triangulation process are found.

The positions of the two markers are automatically found in the recordings using color thresholding and Hough Circle Transform [22]. The recording used for the test consists of 2809 frames where a total of 2739 pair-wise marker detections were found. Care has been taken to move the calibration rod around the entire volume of the aquarium as it is expected that the inter-distance error can depend on the location of the objects for some of the approaches. Inaccuracies may be introduced due to the automatic detection of the markers, however, as the same detections are used for testing all three approaches it is assumed that it does not have any significance.

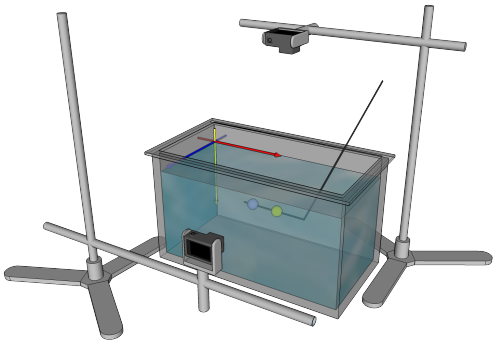


Fig. A.5: Illustration of the test setup and calibration rod with the two colored balls. The setup is used during evaluation of the different approaches. The red, blue, and yellow axis denotes the x-, y-, and z-axis, respectively.

A refractive index of $n_1 = 1.0$ for air and $n_2 = 1.33$ for water is used when calculating the refracted rays in the test. Due to the thickness of the aquarium plastic being only 3mm the refraction caused by the plastic is deemed insignificant and therefore ignored.

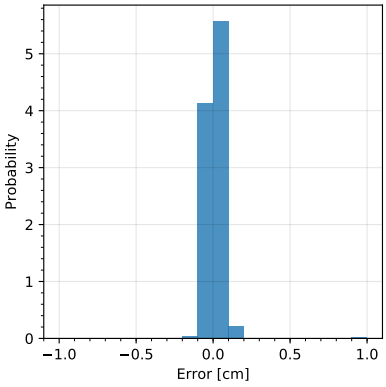
3.1 Results

Histograms of the inter-distance error for the three approaches are shown in Figure A.6. The errors appear to be normally distributed for all three, with the ray tracing based approach performing the best. The results have been gathered in Table A.1.

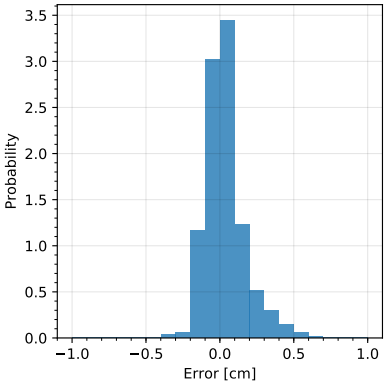
	In air	**Under water**	**Ray tracing**
Mean error (μ)	0.50 cm	0.03 cm	0.01 cm
Deviation (σ)	0.18 cm	0.14 cm	0.09 cm

Table A.1: Results of the 3D reconstruction test.

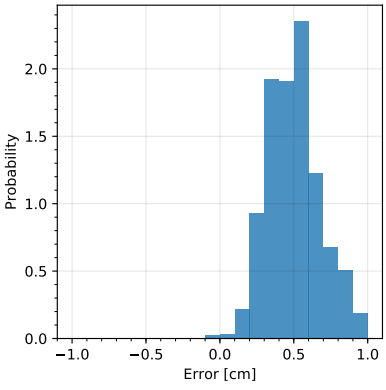
The distribution of the inter-distance error with regard to the location of the markers is illustrated in Figure A.7. Each of the 2739 dots represents the centerpoint between the reconstructed positions of the two markers in the aquarium and the color represents the magnitude of the error, i.e., the absolute inter-distance error.



(a) Ray tracing approach (μ = 0.01 cm, σ = 0.09 cm)

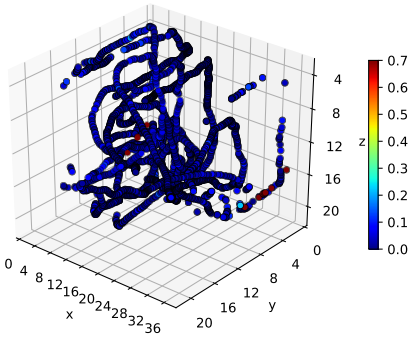


(b) Underwater calibration (μ = 0.03 cm, σ = 0.14 cm)

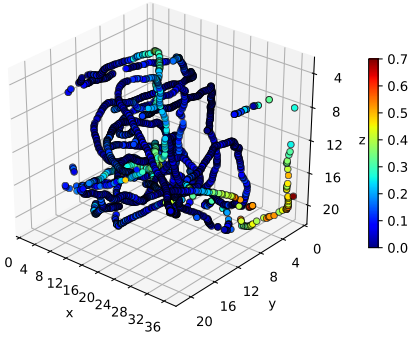


(c) In-air calibration (μ = 0.50 cm, σ = 0.18 cm)

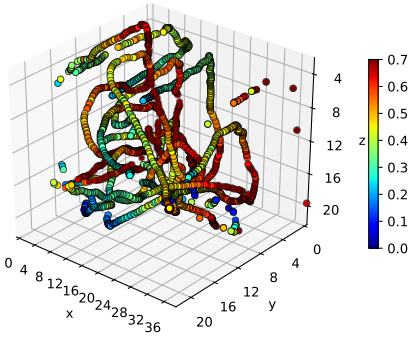
Fig. A.6: Histogram of the inter-distance error for each of the approaches.



(a) Ray tracing approach.



(b) Underwater calibration.



(c) In-air calibration.

Fig. A.7: Plot of the inter-distance error for each of the approaches. The colorbars depict the absolute inter-distance errors in centimeters.

4 Discussion

The in-air camera calibration did not perform well in the conducted 3D reconstruction test, which is not surprising as it has no way of accounting for refraction. This is made apparent in the results as the inter-distance error varies to a large degree based on the location of the markers relative to the cameras, as seen in Figure A.7.

The same is true for the underwater camera calibration approach, but to a lower extend. The degree of deviation of this approach rely on how well the calibration frame, in this case the checkerboard, covers the space of the container. In order to cover the entire container, the checkerboard must be moved around in the aquarium and this may cause disturbances in the water surface that can lead to a further decrease in accuracy and thereby makes it prone to errors. The only approach that seems independent of the location of the markers is the ray tracing based method, with the exception of a few outliers, which is most likely caused by noisy detections such as reflections.

The difference in mean error between the ray tracing based approach and the underwater camera calibration may be attributed to the precision of the measurement of the actual distance between the two markers on the calibration rod and the precision of the detections. These results are hence deemed insufficient to recommend the ray tracing approach over the underwater calibration approach, and vice versa.

The thing that really separates the two approaches is the degree of flexibility, where the ray tracing approach is superior to the under water calibration approach. This is mainly due to the decoupling of the extrinsic and intrinsic parameters, as it allows for a one-time only calibration of the intrinsic parameters. In other words, if the cameras are moved or the aquarium is replaced with another, it is only the extrinsic parameters that need to be estimated, which is easily done by manually annotating the four aquarium corners.

On the other hand, an entire re-calibration with a checkerboard is required for the underwater calibration method if a single feature is changed in the setup. This quickly becomes tedious and time consuming and may, furthermore, create issues with repeatability. This could to some extend be circumvented by replacing the checkerboard with a 3D frame with markers. However,

	In air	**Under water**	**Ray tracing**
Precision	✓	✓✓✓	✓✓✓
Flexibility	✓	✓	✓✓✓
Ease of use	✓✓✓	✓	✓✓

Table A.2: Simplified recap and comparison of the properties of the tested 3D reconstruction approaches.

such a frame would have to be manufactured to the specific tank making the setup even less flexible and impractical for larger setups.

To sum up the findings, the pros and cons of the tested approaches are presented in Table A.2. Precision describes how well the respective approach is capable of reconstructing 3D positions independent of their location in the aquarium. Flexibility is an expression of its capabilities to adapt to variations in the setup, such as change in water levels, camera positions or aquarium size. Lastly, ease of use is an indication for how effortless the calibration of the cameras can be done.

5 Conclusion

The commonly used SVP (single viewpoint) camera model is not applicable when capturing underwater objects due to the refraction of light at the interface between air and water which invalidates the assumption of a single viewpoint. This observation has been confirmed during the test with the in-air camera calibration where the 3D reconstruction of underwater positions showed to be imprecise when refraction was not taken into account at all.

If the SVP camera model is calibrated with a checkerboard, placed under water, a higher precision is obtained as the intrinsic camera parameters can offset the error to some extend. However, this is problematic as even minor changes in the test setup will require the entire system to be re-calibrated, making this approach tedious to use.

Another approach is to use ray tracing in combination with Snell's law to model how the light is refracted and counteract the effect. Tests showed that this approach achieved the lowest mean error and deviation when estimating the position of underwater objects. Furthermore, it has a less restrictive calibration procedure, which makes it less prone to errors and more flexible than the other approaches, why it is recommended to use within the field of automated behavioral analysis of marine animals.

A description of the ray tracing based method is provided, along with a Python implementation found at www.bitbucket.org/aauvap/underwater-camera-calibration. Future work could be to extend the ray tracing method to account for the refraction occurring due to light passing through the material of the aquarium. One solution could be to perform the refraction calculation twice if both the thickness and refractive index of the material are known.

References

[1] A. L. Andrady, "Microplastics in the marine environment," *Marine Pollution Bulletin*, vol. 62, no. 8, pp. 1596–1605, 2011.

[2] M. A. Browne, T. Galloway, and R. Thompson, "Microplastic â€" an emerging contaminant of potential concern?" *Integrated Environmental Assessment and Management*, 2007.

[3] A. Pérez-Escudero, J. Vicente-Page, R. C. Hinz, S. Arganda, and G. G. De Polavieja, "idtracker: tracking individuals in a group by automatic identification of unmarked animals," *Nature methods*, vol. 11, no. 7, pp. 743–748, 2014.

[4] Z.-M. Qian, X. E. Cheng, and Y. Q. Chen, "Automatically detect and track multiple fish swimming in shallow water with frequent occlusion," *PLOS ONE*, vol. 9, no. 9, pp. 1–12, 2014.

[5] S. Macrí, D. Neri, T. Ruberto, V. Mwaffo, S. Butail, and M. Porfiri, "Three-dimensional scoring of zebrafish behavior unveils biological phenomena hidden by two-dimensional analyses," *Nature*, vol. 7, no. 1, may 2017.

[6] A. Yamashita, E. Hayashimoto, T. Kaneko, and Y. Kawata, "3-d measurement of objects in a cylindrical glass water tank with a laser range finder," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, 2003, pp. 1578–1583.

[7] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 22, no. 11, pp. 1330–1334, 2000.

[8] G. Bianco, M. T. Ekvall, J. Bäckman, and L.-A. Hansson, "Plankton 3d tracking: the importance of camera calibration in stereo computer vision systems," *Limnology and Oceanography: Methods*, vol. 11, no. 5, pp. 278–286, 2013.

[9] Z.-M. Qian and Y. Q. Chen, "Feature point based 3D tracking of multiple fish from multi-view images," *PLOS ONE*, vol. 12, no. 6, pp. 1–18, 2017.

[10] E. Simetti, F. Wanderlingh, S. Torelli, M. Bibuli, A. Odetti, G. Bruzzone, D. L. Rizzini, J. Aleotti, G. Palli, L. Moriello, and U. Scarcia, "Autonomous underwater intervention: Experimental results of the maris project," *IEEE Journal of Oceanic Engineering*, vol. PP, no. 99, pp. 1–20, 2017.

[11] G. Bianco, A. Gallo, F. Bruno, and M. Muzzupappa, "A comparative analysis between active and passive techniques for underwater 3d reconstruction of close-range objects," *Sensors*, vol. 13, no. 8, pp. 11 007–11 031, 2013.

[12] L. Kang, L. Wu, and Y.-H. Yang, "Experimental study of the influence of refraction on underwater three-dimensional reconstruction using the svp camera model," *Appl. Opt.*, vol. 51, no. 31, pp. 7591–7603, 2012.

[13] R. Kawahara, S. Nobuhara, and T. Matsuyama, "Dynamic 3d capture of swimming fish by underwater active stereo," *Methods in Oceanography*, vol. 17, pp. 118 – 137, 2016.

[14] Y. Kwon and J. B. Casebolt, "Effects of light refraction on the accuracy of camera calibration and reconstruction in underwater motion analysis," *Sports Biomechanics*, 2006.

[15] K. Müller, J. Schlemper, L. Kuhnert, and K. D. Kuhnert, "Calibration and 3D ground truth data generation with orthogonal camera-setup and refraction compensation for aquaria in real-time," in *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, vol. 3, Jan. 2014, pp. 626–634.

[16] S. Henrion, C. W. Spoor, R. P. M. Pieters, U. K. Müller, and J. L. van Leeuwen, "Refraction corrected calibration for aquatic locomotion research: application of snell's law improves spatial accuracy," *Bioinspiration and Biomimetics*, vol. 10, no. 4, 2015.

[17] P. D. V. Buschinelli, G. Matos, T. Pinto, and A. Albertazzi, "Underwater 3d shape measurement using inverse triangulation through two flat refractive surfaces," in *OCEANS 2016 MTS/IEEE Monterey*, 2016, pp. 1–7.

[18] A. Agrawal, S. Ramalingam, Y. Taguchi, and V. Chari, "A theory of multi-layer flat refractive geometry," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[19] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[20] *An Introduction to Ray Tracing*. London, UK: Academic Press Ltd., 1989.

[21] R. I. Hartley and P. Sturm, "Triangulation," *Computer Vision and Image Understanding*, vol. 68, no. 2, pp. 146–157, 1997.

[22] H. K. Yuen, J. Princen, J. Dlingworth, and J. Kittler, "A comparative study of hough transform methods for circle finding," in *Proceedings of the Alvey Vision Conference*, 1989.

Paper B

3D-ZeF: A 3D Zebrafish Tracking Benchmark Dataset

Malte Pedersen, Joakim Bruslund Haurum,
Stefan Hein Bengtson, and Thomas B. Moeslund

The paper has been published in the
Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2426-2436, 2020.

Malte Pedersen, Joakim Bruslund Haurum, Stefan Hein Bengtson, and Thomas B. Moeslund, ”3D-ZeF: A 3D Zebrafish Tracking Benchmark Dataset. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

The layout has been revised.

Abstract

In this work we present a novel publicly available stereo based 3D RGB dataset for multi-object zebrafish tracking, called 3D-ZeF. Zebrafish is an increasingly popular model organism used for studying neurological disorders, drug addiction, and more. Behavioral analysis is often a critical part of such research. However, visual similarity, occlusion, and erratic movement of the zebrafish makes robust 3D tracking a challenging and unsolved problem.

The proposed dataset consists of eight sequences with a duration between 15-120 seconds and 1-10 free moving zebrafish. The videos have been annotated with a total of 86,400 points and bounding boxes. Furthermore, we present a complexity score and a novel open-source modular baseline system for 3D tracking of zebrafish. The performance of the system is measured with respect to two detectors: a naive approach and a Faster R-CNN based fish head detector. The system reaches a MOTA of up to 77.6%. Links to the code and dataset is available at the project page
`http://vap.aau.dk/3d-zef`

1 Introduction

Over the past decades, the use of zebrafish (*Danio rerio*) as an animal model has increased significantly due to its applicability within large-scale genetic screening [1, 2]. The zebrafish has been used as a model for studying human neurological disorders, drug addiction, social anxiety disorders, and more [3–8]. Locomotion and behavioral analysis are often critical parts of neuroscientific and biological research, which have traditionally been conducted manually [9–11]. However, manual inspection is subjective and limited to small-scale experiments. Therefore, tracking systems are getting increasingly popular due to their efficiency and objectivity. The majority of the solutions has been developed for terrestrial animals or fish in shallow water, and most studies have been based on 2D observations in scientific [12–18] and commercial systems [19–22]. However, observations in a single plane cannot capture all the relevant phenotypes of fish [23–25]. Estimating the 3D trajectories of multiple zebrafish accurately is difficult due to their erratic movement, visual similarity, and social behavior [26], see Figure B.1. This may be one of the reasons why no commercial solution has been developed yet. Only few groups in the scientific community have addressed the problem, focusing mainly on stereo vision [27–31] and monocular stereo using mirrors [32, 33]. However, no labeled datasets have been made publicly available within the field, which makes a fair comparison between the applied methods difficult. This ultimately hinders significant developments in the field as we have seen in other computer vision fields with common datasets.

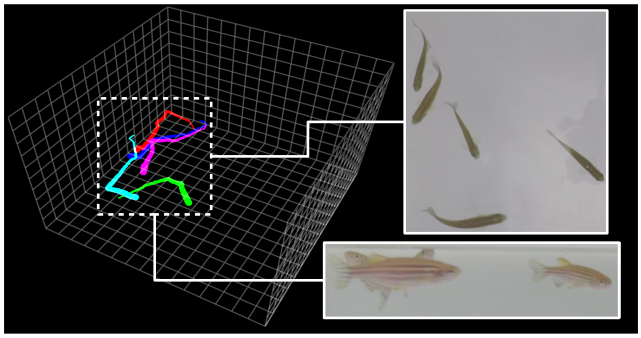


Fig. B.1: An example that illustrates the difference between the two perspectives. The 3D trajectories are estimated based on the head point annotations.

Therefore, our contributions are

- a publicly available RGB 3D video dataset of zebrafish with 86,400 bounding box and point annotations.
- an open-source modular baseline system.

A large part of 3D multi-object tracking methods are developed for LiDAR-based traffic datasets [34–38] or RGB-D tracking [39, 40]. However, to the best of our knowledge, there exists no publicly available annotated RGB stereo dataset with erratic moving and similarly looking subjects like the one we propose.

2 Related Work

Multi-Object Tracking (MOT). Reliably tracking multiple objects is widely regarded as incredibly difficult. The interest in solving MOT has been steadily increasing since 2015 with the release of the MOT [41–43], UA-DETRAC [44, 45], and KITTI [34, 35] challenges. Within the MOT challenges, the current focus is on either aiming to solve the association problem using deep learning [46], using techniques such as intersection-over-union based tracking [47], or disregarding tracking-specific models and utilizing the improvements within object detections [48].

Zebrafish Tracking. Vision-based tracking systems developed for studying animal behavior have traditionally been based on 2D [18, 49–54] due to simplicity and because the movement of most terrestrial animals can be approximated to a single plane. The majority of research in zebrafish tracking has followed this path by only allowing the fish to move in shallow water and assuming that motion happens in a 2D plane.

A 2D animal tracker, called idTracker presented by Perez-Escudero et al. in 2014 [49], uses thresholding to segment blobs and is able to distinguish between

individual zebrafish based on intensity and contrast maps. In 2019, Romero-Ferrero et al. presented an updated version of idTracker, called idtracker.ai [18], which is the current state-of-the-art 2D tracker system based on convolutional neural networks (CNN) for handling occlusions and identifying individuals. The subjects are observed with a camera positioned above a tank with a water depth of 2.5 cm and the distance between camera and subjects is, therefore, approximately the same at all times. As stated by the authors, this simplifies the task compared to a real 3D tracking scenario.

However, as most aquatic species move in three dimensions, trajectories in 3D are required to thoroughly describe their behavior [55, 56]. The most frequently used acquisition method when dealing with studies of animal behavior in 3D is stereo vision [28, 30, 31, 56–61]. 3D tracking of zebrafish has been focused mainly on single subjects or small groups, as occlusion is a big hindrance for maintaining correct IDs due to their shoaling behavior [26]. Furthermore, the visual appearance of the fish can change dramatically depending on the position and posture, which makes re-identification more complex compared to 2D.

The Track3D module from the commercial EthoVision XT [19] is popular for tracking zebrafish in 3D, but is limited to a single individual [56, 61]. An early semi-automatic 3D tracking system was developed by Viscido et al. [58] to investigate the relationship between individual members of fish schools. Initial 2D tracks were generated by a nearest neighbor algorithm followed by a step allowing the user to adjust and correct the proposed 2D trajectories, and subsequently triangulated to reconstruct the 3D trajectories.

Qian et al. have worked extensively with tracking of zebrafish and have developed a 2D tracking system with a top-view camera using an augmented fast marching method (AFMM) [62] and the determinant of the Hessian [15]. This was expanded to 3D tracking by extending the setup with a side-view camera. AFMM was utilized to generate a feature point based fish representation in each view followed by 2D tracklet construction based on motion constraints. 3D tracks were then constructed by associating the 2D tracklets with side-view detections using epipolar and motion consistency constraints [29]. Liu et al. [63] extended this method to better handle occlusions based on a set of heuristic methods and the epipolar constraint. A third camera was added in [31], and the feature point representation method was extended.

Cheng et al. [28] utilized a similar three-camera setup, applying an iterative unsupervised learning method to train a CNN-based classifier to distinguish between the individual fish from a camera placed above the water tank. The classifier was trained on the head region of the fish during periods when all fish were visible at the same time. By iteratively retraining the classifier, they were able to generate 2D tracks from the top-view and reconstruct the 3D tracklets based on detections from the two other side-view cameras under epipolar and motion constraints.

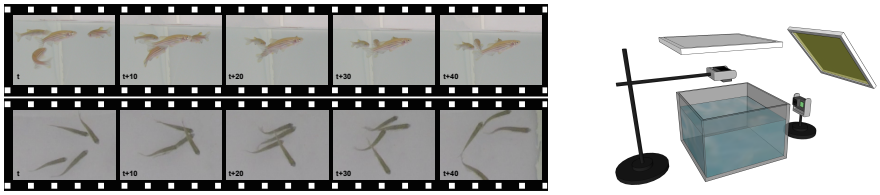


Fig. B.2: Five frames from two different occlusion scenarios. The upper frames are from the front-view and the lower frames are from the top-view. An illustration of the experimental setup is shown to the right.

Wang et al. [30] also utilized a three-camera setup, using a Gaussian Mixture Model, a Gabor filter and an SVM-based method to detect the fish heads in the top- and side-views, respectively. The top-view detections are associated into 2D tracklets based on a cross-correlation method and by applying a Kalman filter; near linear movement is achieved by a frame rate of 100 FPS. The 2D tracklets are then constructed into 3D tracklets by associating the side-view detections under epipolar and motion constraints. In [64], Wang et al. proposed to model the top-view movement of the zebrafish through long short-term memory networks, which were used to improve the motion constraints in a new iteration of their 3D system [65]. Lastly, Wang et al. used a CNN for re-identification of zebrafish heads from the top-view [66], although this has yet to be incorporated into a 3D tracking setup. None of the methods are able to track multiple zebrafish in 3D for more than a few seconds without ID swaps; this is still a difficult and unsolved problem.

Datasets. As in other MOT challenges, there is a mutual agreement that occlusion is what makes 3D tracking of zebrafish difficult. Nonetheless, only Wang et al. [65] describe their recordings based on occlusion frequency; however, they do not define how it is measured. Qian et al. [31] indicate their complexity based on the amount of fish, but only four occlusion events occur during their 15 seconds demo video with ten fish. For comparison, there are 66 occlusion events in our 15 seconds sequence with ten fish.

3 Proposed Dataset

The proposed 3D zebrafish dataset, 3D-ZeF, has been recorded from a top- and front-view perspective. This approach was taken to minimize events of total occlusion typical for side-by-side binocular setups. An example of the visual variation between the views is shown in Figure B.2 together with an illustration of the experimental setup.

3.1 Experimental Setup

The setup used to record the proposed dataset was built entirely from off-the-shelf hardware, whereas previous methods have used specialized camera equipment. An illustration of the setup is shown in Figure B.2. The two light panels are IKEA FLOALT of size 30×30 cm with a luminous flux of 670 lumen and a color temperature of 4000K. The test tank is a standard glass aquarium of size $30 \times 30 \times 30$ cm with a water depth of 15 cm. The top and front cameras are GoPro Hero 5 and GoPro Hero 7, respectively. All the videos are recorded with a resolution of 2704×1520, 60 FPS, $1/60$ s shutter speed, 400 ISO, and a linear field of view. However, the fish tank does not take up the entire image, therefore, the effective region of interest is approximately 1200×1200 and 1800×900 for the top- and front-view, respectively. Diffusion fabric was placed in front of the top light in order to reduce the amount of glare in the top-view. Semi-transparent plastic was attached to three out of four of the window panes in order to reduce reflections. Furthermore, the front camera was placed orthogonally to the water level, which reduced reflections from the water surface. Lastly, the pair-wise recordings have been manually synchronized using a flashing LED, which results in a worst case temporal shift of $\frac{1}{2 \cdot \mathrm{FPS}}$.

3.2 Dataset Construction

A total of eight sequences were recorded and divided into a training, validation, and test split. Each sequence consists of a pair of temporally aligned top- and front-view videos and the specifications of the three splits are shown in Table B.1. In order to avoid data leakage, each split contains a unique set of fish. The training and validation set of fish were from the same cohort, whereas the fish in the test split were from a younger cohort. Therefore, the test set differs from the training and validation set, as the fish are smaller and behave socially different. This represent a real-life scenario where different cohorts need to be tracked, which has not generally been addressed within the field.

The zebrafish were manually bounding box and point annotated with consistent identity tags through all frames. The bounding boxes were tightly fitted to the visible parts of the zebrafish and the point annotations were centered on the head. If a set of fish touched, an occlusion tag was set for all involved bounding boxes. During occlusions, the bounding box was fitted to the visible parts of the fish and not where it was expected to be due to the extreme flexibility of the zebrafish. The pair-wise point annotations from the two views were triangulated into 3D positions using the method proposed by Pedersen et al. [67]. The fish head was approximated during occlusions to ensure continuous 3D tracks.

It should be noted that the data was recorded in RGB. Zebrafish can change

	Trn2	Trn5	Val2	Val5	Tst1	Tst2	Tst5	Tst10	Total
Length	120 s	15 s	30 s	15 s	15 s	15 s	15 s	15 s	240 s
Frames	14,400	1,800	3,600	1,800	1,800	1,800	1,800	1,800	28,800
BBs	28,800	9,000	7,200	9,000	1,800	3,600	9,000	18,000	86,400
Points	28,800	9,000	7,200	9,000	1,800	3,600	9,000	18,000	86,400
OC	1.82 / 1.42	3.60 / 2.93	0.93 / 0.47	2.67 / 3.80	0.00 / 0.00	0.67 / 0.67	3.07 / 2.93	4.40 / 6.53	
OL	0.41 / 0.51	0.56 / 0.64	0.22 / 0.63	0.25 / 0.66	0.00 / 0.00	0.10 / 0.38	0.25 / 0.36	0.28 / 0.35	
TBO	0.69 / 0.89	1.00 / 1.21	1.79 / 3.20	1.64 / 0.73	15.00 / 15.00	2.41 / 2.18	1.38 / 1.28	1.86 / 1.40	
IBO	0.29 / 0.26	0.28 / 0.28	0.24 / 0.35	0.22 / 0.34	0.00 / 0.00	0.19 / 0.19	0.25 / 0.23	0.26 / 0.24	
Ψ	**0.26**	**0.50**	**0.03**	**0.63**	**0.00**	**0.01**	**0.16**	**0.28**	

Table B.1: Overview of the proposed dataset. OC, OL, TBO, and IBO are listed for the top- and front-view, respectively, and the number of fish is denoted in the sequence name. OC: average amount of occlusions per second, OL: average occlusion length in seconds, TBO: average amount of seconds between occlusions, IBO: intersection between occlusions, Ψ: complexity measure based on OC, OL, TBO and IBO (see Equation (B.2)).

their body pigmentation based on their environment, stress level, and more [23]. The changes in coloration can be important in behavioral studies and may even be valuable in solving the 3D tracking problem.

3.3 Dataset Complexity

Intuitively, a higher number of fish creates a more difficult tracking problem. However, this is only true to some extent as the main complexity factor is the number and level of occlusions, which depends on a combination of the social activity and amount of space rather than the number of individuals. Therefore, we have defined a range of metrics based on occlusion events to describe the complexity of the proposed sequences. An occlusion event is defined by a set of consecutive frames, where a fish is part of an occlusion. The events are measured from the perspective of the fish; if two fish are part of an occlusion it counts as two events.

The number of occlusion events indicates how often a fish is part of an occlusion, but, few long occlusions can be just as problematic as many short. The length of the occlusions and time between them are, therefore, important to keep in mind when evaluating the complexity of a recording. Due to our definition of occlusion events there are cases where fish are part of occlusions with only minor parts of their bodies. Therefore, the intersection between occlusions is measured as an indication of the general intersection level. The metrics that we provide as basis for the complexity level of our recordings are defined here:

Occlusion Count (OC): the average number of occlusion events per second.
Occlusion Length (OL): the average time in seconds of all occlusion events.
Time Between Occlusions (TBO): the average time in seconds between occlusion events.
Intersection Between Occlusions (IBO): a measure of how large a part of the fish that is part of an occlusion event. The intersection in a frame, f, for fish i

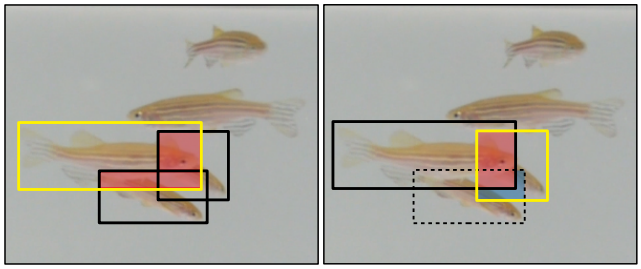


Fig. B.3: IBO seen from the perspective of two different individuals in the same frame. The targets are marked in yellow, the red area shows the intersection with a subject that is part of the same occlusion as the target, and the blue area shows the intersection with a subject that is not part of the same occlusion as the target.

is given by

$$\mathrm{IBO}_{i,f} = \frac{1}{|\mathrm{bb}_i|} \sum_{j=1}^{n_{\mathrm{occ}}} \mathrm{bb}_i \cap \mathrm{bb}_j, \quad \text{for } j \neq i, \tag{B.1}$$

where n_{occ} is the number of fish in an occlusion event, and bb_j is the set of pixel coordinates in the bounding box of fish j. IBO is measured across all bounding boxes with an occlusion tag in a given frame, even for subjects that are not part of the same occlusion. Two examples are presented in Figure B.3, where the $\mathrm{IBO}_{i,f}$ is calculated from the perspective of the targets enclosed in yellow. The blue area in the second example, represents the intersection with a subject that is not part of the same occlusion as the target. Additionally, the annotated bounding boxes enclose only the visible parts of the subjects. Thus, the actual intersection between the subjects can be higher if a large part of a fish is hidden. Nonetheless, the assumption is that a high IBO is an expression of heavy occlusion and vice versa. The IBO measure presented in Table B.1 is an average between all fish in all frames. A single complexity measure is calculated per sequence, by combining the four proposed metrics by

$$\Psi = \frac{1}{n} \sum_{v}^{\{\mathrm{T,F}\}} \frac{\mathrm{OC}_v \, \mathrm{OL}_v \, \mathrm{IBO}_v}{\mathrm{TBO}_v}, \tag{B.2}$$

where n is the number of camera views and subscript T and F denote the top- and front-view, respectively. If a recording has no occlusions the complexity measure, Ψ, is zero; otherwise, the measure is in the interval $]0, \infty[$, where a larger value indicates a higher complexity.

4 Method

The pipeline of the proposed 3D tracker follows a modular tracking-reconstruction approach, where subjects are detected and tracked in each view before being triangulated and associated across views. This allows us to use the temporal information of the tracklets in the two views in the 3D association step in opposition to a reconstruction-tracking approach, where detections are triangulated before tracks are generated.

4.1 Object Detection in 2D

A consistent 2D point is needed in each view in order to create 3D trajectories. As the head is the only rigid part of the body the tracking point is chosen to be located between the eyes of the fish. We present two simple methods to find the head-point of the fish: a naive approach, that does not require training, and a CNN based approach.

Naive: A background image, bg, is initially estimated for each view by taking the median of N_{bg} images sampled uniformly across the videos. Subsequently, the background is subtracted by calculating the absolute difference image, $fg = |im - bg|$. To locate the head of a fish in the top-view, the fg is binarized using the intermodes bimodal threshold algorithm [68]. The skeletonization approach of Zhang and Suen [69] is applied, and the endpoints are analyzed to locate the head of the fish. In the front-view the fg is binarized through the use of a histogram entropy thresholding method because the appearance of the fish cannot be approximated as bimodal. The head point is estimated as being either the center of the blob or one of the middle edge points of the blob along the minor axis of the detected bounding box. All three points are evaluated during the 3D reconstruction step, and the two points with the highest reprojection errors are discarded.

FRCNN-H: A Faster R-CNN [70] model has been trained for each view. The bounding boxes have been extracted from all the head-point annotations in the training sequences in order to train a head-detector model for each view. The bounding boxes have static diameters of 25 and 50 pixels for the top-, and front-view, respectively. The head-points are determined as the center of the detected bounding boxes which have a minimum confidence of c.

See the supplementary material for more detailed information on the detectors.

4.2 2D Tracklet Construction

As zebrafish move erratically, it is difficult to set up a stable motion model. Therefore, we use a naive tracking-by-detection approach. The tracking is done by constructing a distance matrix between the detections in a frame and the

last detections of current tracklets. The matrix is solved as a global optimization problem using the Hungarian algorithm [71]. Tracklets are deliberately constructed in a conservative manner, where robustness is encouraged above length. A new detection is only assigned to a tracklet located within a minimum distance, denoted δ_{T} and δ_{F}, for the top and front view respectively. If a tracklet has not been assigned a detection within a given amount of time, τ_k, the tracklet is terminated.

The ℓ_2 distance between the head detections is used in both views for the FRCNN-H method. However, the Mahalanobis distance between the center-of-mass is used for the front-view in the Naive method. This is due to the elliptical form of the zebrafish body, which can be utilized by setting the covariance matrix of the blob as the Mahalanobis matrix; as the fish is more likely to move along the major axis than along the minor axis.

4.3 2D Tracklet Association Between Views

The 2D tracklets from each view are associated into 3D tracklets through a graph-based approach. All 2D tracklets with less than a given number of detections, α, are removed in order to filter out noisy tracklets. The 3D calibration and triangulation method from Pedersen et al. [67] is used.

Graph Construction

A directed acyclic graph (DAG) is constructed. Every node represents a 3D tracklet and consists of two 2D tracklets; one from each camera view. Each edge associates nodes, where the 3D tracklet is based on the same 2D tracklet from one of the views.

Create nodes: The graph nodes are constructed by processing each top-view tracklet and identifying all temporally intersecting front-view tracklets as given by

$$I = F_{\mathrm{T}} \cap F_{\mathrm{F}}, \tag{B.3}$$

where F_{T} and F_{F} are the set of frames with detections in the top- and front-view tracklets, respectively, and I is the set of frames with detections in both views. If $I = \emptyset$, the node is not created.

An example is presented in Figure B.4, where both the blue and red tracklets in the top-view intersects with the three tracklets in the front-view. The outer and inner circles of the six nodes represent the top- and front-view tracklets, respectively. The number inside the nodes indicates the node weight, which is calculated as follows.

For each intersecting frame in I, denoted f, the 2D tracklets are triangulated. This results in a 3D point of the zebrafish head, p_f, with a reprojection error, x_f. For the Naive method where the head is not directly detected in the front-view, the top-view 2D point is triangulated with the three estimated

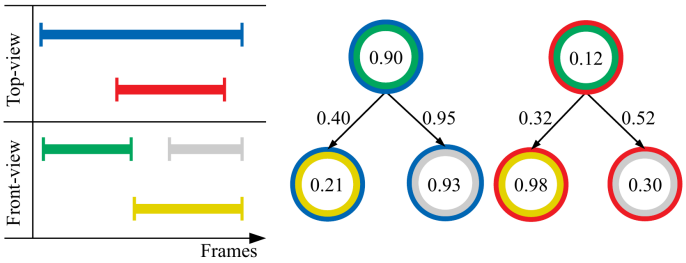


Fig. B.4: The colored lines represent 2D tracklets in each view, the node pairs are represented by the double-colored circles, and the edges of the DAG are shown by the arrows. The numbers represent example node and edge weights.

points to find the match resulting in the smallest reprojection error. Therefore, p_f represents the point with the smallest reprojection error. To penalize large reprojection errors, the complimentary probability from the exponential cumulative distribution function (CDF), Φ, is utilized. The exponential CDF is chosen as it approximately models the reprojection error of the ground truth training data. The set of weights for all valid 3D points, V, can be described by the following set-builder notation

$$V = \{1 - \Phi(x_f \mid \lambda_{err}) \mid f \in I \wedge A(p_f)\}, \tag{B.4}$$

where λ_{err} is the reciprocal of the mean of the training data reprojection error, and A states whether p_f is within the water tank. The per-frame weights in V are combined into a single weight, W, for the entire node by

$$W = \text{median}(V)\frac{|V|}{|F_\text{T} \cup F_\text{F}|}, \tag{B.5}$$

and the node is added to the DAG given that $W \neq 0$. This weighting scheme considers both the reprojection error and the ratio of frames with valid 3D points compared to the set of all frames I. The median function is used instead of the mean function in order to counteract that a few extreme outliers skew the weight.

Connect nodes: The nodes in the DAG should be connected to all other nodes building on one of the same 2D tracklets, as long as the 2D tracklets in the other view do not overlap temporally, as illustrated in Figure B.4. This is done by constructing the set of node pairs, P, from the set of nodes in the DAG, N. Each element of N, denoted n, consists of the 2D tracklets, t_F and t_T, the 3D tracklet, t, and the node weight, W. Nodes n_i and n_j are considered a pair if $t_{i,\text{T}} = t_{j,\text{T}}$ or $t_{i,\text{F}} = t_{j,\text{F}}$, if the 2D tracklets in the other view do not temporally overlap, and if t_i starts earlier in time than t_j. This is necessary in order to avoid assigning multiple detections to the same frame.

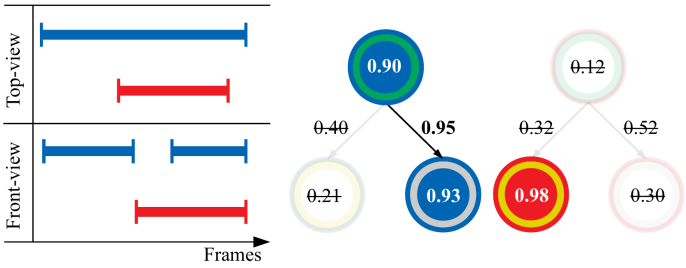


Fig. B.5: Graph evaluation based on the example from Figure B.4. The colored lines represent 2D tracklet pairs based on the chosen nodes in the graph; the transparent nodes are discarded.

This can be represented by the set-builder notation

$$P = \{(n_i, n_j) \mid n_i, n_j \in N \wedge O(n_i, n_j) \wedge T(n_i, n_j)\}, \tag{B.6}$$

where O assesses whether t_i starts before t_j, and T ensures that the 2D tracklets in n_i and n_j do not temporally overlap, where $n = \{t_\mathrm{T}, t_\mathrm{F}, t, W\}$.

For each node pair in P, the weight, E, of the directed edge from n_i to n_j is based on:

- s, the speed of the fish as it moves between the last detection in t_i and the first detection in t_j.
- t_d, the temporal difference between t_i and t_j.
- W_i and W_j, the weights of the nodes.

The edge weight is calculated as the complimentary probability of the CDF of the exponential distribution, Φ. The exponential distribution is chosen as it approximately models that of the speed of the zebrafish. E is calculated by

$$E = (1 - \Phi(s \mid \lambda_s))e^{-\frac{t_d}{\tau_p}}(W_i + W_j), \tag{B.7}$$

where τ_p is an empirically chosen value, and λ_s is the reciprocal of the sum of the mean and standard deviation of the measured speed in the training data. In case a node is not present in any node pairs, the node will be assigned to the DAG, but it will have no edges. The DAG is therefore a disconnected graph.

Graph Evaluation

The final 3D tracklets are extracted from the constructed DAG; this is done by recursively finding the longest path in the graph and storing the set of nodes as a single 3D tracklet. The longest path is the path throughout the DAG, which gives the highest value when summing all nodes and edge weights in the path, see Figure B.5. After extraction of a path, the used nodes, and all

other nodes using the same 2D tracklets, are removed from the DAG. This process is repeated until the DAG is empty. In case a 2D tracklet in the 3D tracklet is missing a detection, the 3D position cannot be assigned, but the known information of the 2D tracklet is kept. For the Naive method, the head position of the front-view 2D tracklet is determined by assigning the estimated point, which minimizes the ℓ_2 distance to the head positions in the consecutive frame.

4.4 3D Tracklet Association

The final 3D tracks are constructed from the 3D tracklets in a greedy manner. A set of tracklets equal to the amount of fish present, N_{fish}, is used as initial *main tracklets*. The remaining tracklets, denoted *gallery tracklets*, are assigned one by one to a single main tracklet, until no more tracklets can be assigned.

Initial Tracklet Selection

The set of N_{fish} in the main tracks is selected by finding the stable tracklets that are temporally concurrent in time and span long time intervals. For each tracklet, the set of other temporally concurrent tracklets is considered. In this set, all possible combinations of size N_{fish} are investigated. If all tracklets in the set overlap temporally, the set is saved as a valid tracklet set. The valid tracklet set with the highest median temporal overlap is used to construct N_{fish} full 3D tracks. This is done by using the greedy association scheme described in the following sections. No 3D tracks are created if no valid combination of size N_{fish} is identified.

Greedy Association

A greedy association algorithm is used when each gallery tracklet is associated with a single main tracklet. The greedy part of the algorithm concerns the way that gallery tracklets are chosen; all gallery tracks are ranked in ascending order by the shortest temporal distance to any main tracklet. If the gallery tracklet overlaps temporally with all main tracklets, it is relegated to the end of the list. When the gallery tracklet has been associated with a main track, the remaining gallery tracks are re-ranked, and the process repeated. In this way, the main tracklets are "grown" into full tracks. The gallery tracklet assignment is based on minimizing the cost of assignment. The cost is based on a set of distance measures, which are determined from two cases.

In the first case at least one main tracklet does not temporally overlap with the gallery tracklet. In this case, the association process is based on the spatio-temporal distances between the gallery tracklet and main tracklets. All temporally overlapping main tracklets are not considered.

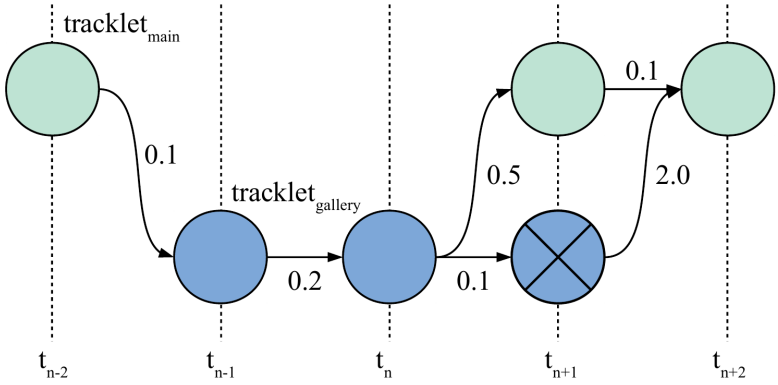


Fig. B.6: Example of internal spatio-temporal DAG, with the spatial distance between detections in the tracklets. The shortest path is found when switching from $\text{tracklet}_{\text{gallery}}$ to $\text{tracklet}_{\text{main}}$ in frame t_{n+1}.

In the second case the gallery tracklet overlaps temporally with all main tracklets. As the spatio-temporal distances between the main and gallery tracklet is no longer measurable, a different set of distance values are used: The internal spatio-temporal distances, the amount of intersecting frames, i.e., frames with a detection in both the main and gallery tracklets, and the ratio of intersecting frames compared to the total amount of detections in the gallery tracklet. The internal spatio-temporal distances are determined through the construction of a DAG, where each node is a detection in a frame, and the edge weights are the spatial distances between the temporally previous nodes. The final path is the one minimizing the spatial distance traveled. An example of a graph is shown in Figure B.6. The distances are calculated as the mean of the values when the graph switches from a detection in the gallery tracklet to the main tracklet and vice versa.

Association: The distance measures are consolidated into a single assignment decision through a global cost scheme. Each distance value is normalized across valid main tracklets into the range $[0; 1]$ and sum to 1. The final cost of assigning the gallery tracklet to a main tracklet, is obtained by calculating the mean of the normalized distance values. The gallery tracklet is associated with the main tracklet with the smallest cost, unless all main tracklet costs are located within a small margin, β, of each other, in which case the gallery tracklet is discarded. β directly enforces a margin of confidence in the assignment, in order to not assign a gallery traklet based on inconclusive cost values.

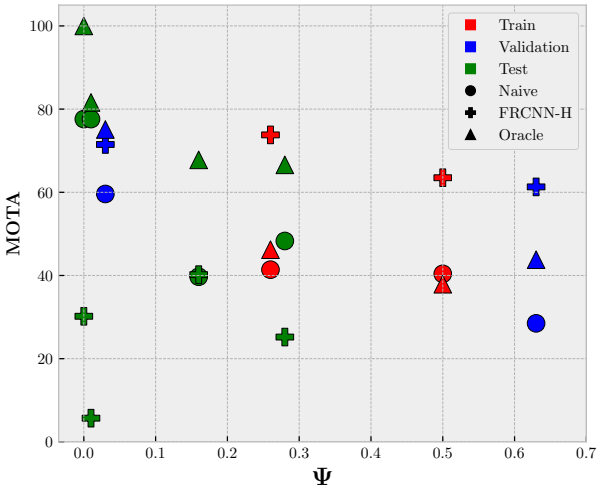


Fig. B.7: MOTA compared to the dataset complexity, Ψ, for all sequences in the dataset.

5 Evaluation

The metrics used in the MOT challenges [41–43] and the Mean Time Between Failures (MTBF) proposed by Carr and Collins [72] are utilized to measure the performance of the system on the proposed dataset. The MOT challenge metrics consist of the CLEAR MOT metrics [73], the mostly tracked/lost metrics [74], and the identification-based metrics [75].

The final 3D tracks are evaluated based on a subset of the MOT challenge metrics and the monotonic MTBF metric. The primary metric used is the multiple object tracking (MOTA) metric. The detected and ground truth tracklets are compared using the detected and annotated head points. A detection is only associated with a ground truth tracklet if it is within a distance of 0.5 cm. The performance of the system is evaluated with two different detection modules: Naive, and FRCNN-H. The results are compared with a hypothetical tracker, called Oracle, which tracks perfectly at all times except during occlusions. This provides an upper bound on the performance if occlusions are not handled in any way. The full set of metrics, system parameters, and results can be found in the supplementary material.

Results for all sequences compared to data complexity is shown in Figure B.7, and metrics for the test sequences are shown in Table B.2. It is clear that the FRCNN-H outperforms the Naive method on the training and validation splits; it even outperforms the Oracle tracker in three out of four cases.

	Method	MOTA ↑	MT ↑	ML ↓	ID Sw. ↓	Frag. ↓	$MTBF_m$ ↑
Tst1	Naive	77.6%	1	0	0	28	12.5
	FRCNN-H	30.2%	0	0	0	15	8.212
	Oracle	100.0%	1	0	0	0	900
Tst2	Naive	77.6%	1	0	0	44	15.856
	FRCNN-H	5.7%	0	2	2	17	2.641
	Oracle	81.6%	2	0	0	25	27.396
Tst5	Naive	39.7%	0	0	7	185	6.249
	FRCNN-H	40.2%	0	0	7	115	7.577
	Oracle	67.8%	1	0	0	50	28.112
Tst10	Naive	48.3%	0	0	11	268	9.075
	FRCNN-H	25.2%	0	3	32	225	4.904
	Oracle	66.6%	1	10	0	119	23.105

Table B.2: Evaluation of 3D tracks on test split. The arrows indicate whether higher or lower values are better. MOTA: Multiple Object Tracking Accuracy, MT: Mostly tracked, ML: Mostly lost, ID Sw.: Number of identity swaps, Frag.: Number of fragments, $MTBF_m$: Monotonic MTBF.

This is likely due to the method being able to detect some of the fish heads during occlusions. However, the superior performance is only seen on the two splits where the fish are from the same cohort. On the test set the FRCNN-H fails to generalize, whereas the Naive method still manages to track the fish.

It should be noted that the poor performance of the Naive method on Tst1, is suspected to be due many short tracks from erratic movement, which the pipeline with the used parameter settings does not handle well.

5.1 Comparison with Other Methods

It has not been possible to make a fair comparison with the other 3D zebrafish tracking methods mentioned in Section 2. Previous systems have been analyzed in terms of ID swaps, fragments, precision, and recall for the generated 2D and 3D tracks. However, there is no exact description of how these metrics are calculated. The evaluation protocol is further limited by not including a statement on the maximum allowed distance between estimated and ground truth tracks leading to uncertainty on the accuracy of the metrics.

Furthermore, the evaluated sequences are not described in terms of complexity, even though occlusion is repeatedly stated as a major hindrance in 3D zebrafish tracking. The only common complexity indication of the datasets is the number of fish, even though it is not representative. An example of this is the tracking demo video of Qian et al. [62] with ten fish and only four occlusion events during 15 seconds. Wang et al. [30] describes their dataset on basis of an occlusion probability but do not explain how it is measured.

There are currently no publicly available annotated data and the previous systems are evaluated on seemingly simplified cases of the problem. Furthermore, the used evaluation protocols are lacking details in such a manner that

it is not possible to determine under which conditions the metrics have been calculated. This, along with inaccessible codebases, severely limits the reproducibility of the results, and makes it impossible to ensure identical evaluation procedures. Therefore, it simply does not make sense to compare the proposed system to the other methods under the current circumstances.

6 Conclusion

Zebrafish is an increasingly popular animal model and behavioral analysis plays a major role in neuroscientific and biological research. However, it is tedious and subjective to manually describe the complex 3D motion of zebrafish. Therefore, 3D zebrafish tracking systems are critically needed to conduct accurate experiments on a grand scale. The significant development experienced in other fields of MOT has not yet translated to 3D zebrafish tracking. The main reason being that no dataset has been made publicly available with ground truth annotations. Therefore, we present the first publicly available RGB 3D zebrafish tracking dataset called 3D-ZeF.

3D-ZeF consists of eight stereo sequences with highly social and similarly looking subjects demonstrating complex and erratic motion patterns in three dimensions that are not seen in common MOT challenges. A complexity measure based on the level of occlusions has been provided for each sequence to make them comparable to future related datasets. The proposed dataset is annotated with 86,400 bounding boxes and points; the latter used for estimating ground truth 3D tracks based on the head position of the fish. Different cohorts of zebrafish are used in the training, validation, and test splits to avoid data leakage; a problem that has never been addressed within the field.

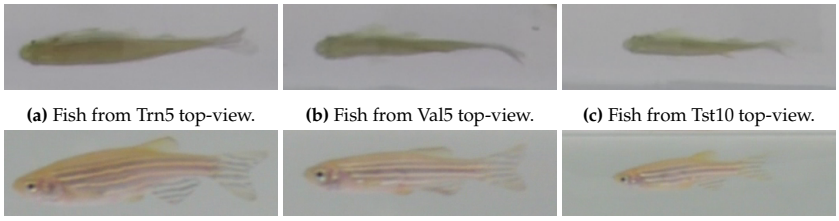
The proposed Naive method scores a MOTA between 25% and 80% across the entire dataset, which correlates well with the complexity measure of the recordings. The open-source modular based system provides a baseline and stepping stone for further development within the field of 3D zebrafish tracking and understanding.

B Supplementary Material

In these supplementary materials we provide: examples of the visual difference between the fish in the three splits, a more detailed explanation of the detectors, details on the benchmark pipeline parameters, a more detailed presentation of the relation between the proposed dataset complexity measures and tracking metrics, and the full set of tracking metrics for each step in the proposed benchmark pipeline.

B.1 Fish Examples

The visual appearance of the fish varies both within and between the splits as unique groups of fish have been used for each split. However, the two groups of fish used in the train and validation sets are from the same cohort, whereas the zebrafish in the test split are from a cohort of younger and smaller fish. We present a fish from each of the three splits in Figure 8 captured at the approximately same position in the water tank. The resolution of the zebrafish from the test split is significantly smaller compared to the fish in the training and validation splits, mainly due to the physical size of the fish.



(a) Fish from Trn5 top-view. **(b)** Fish from Val5 top-view. **(c)** Fish from Tst10 top-view.

(d) Fish from Trn5 front-view. **(e)** Fish from Val5 front-view. **(f)** Fish from Tst10 front-view.

Fig. 8: Examples of zebrafish from the train, validation, and test split captured from the approximately same position in the water tank. The top- and front-view pairs show the same fish. Notice the variation in size of the zebrafish.

B.2 Naive Object Detection

The Naive object detection algorithms presented briefly in the paper are described in more details in this section. The methods are inspired by Qian et al. [62] and their work on tracking zebrafish in 2D.

Pre-Processing

Initially the background image, without any fish, is estimated by taking the median of N images sampled uniformly across the video. In the tests presented

in the paper a set of 80 images were used to create the background image for each recording. All further processing is restricted within a defined region of interest based on the boundaries of the water tank and the level of water, in order to limit the processing time.

The video is downsampled by a factor of 2, in order to further decrease the processing time. Background subtraction is applied by calculating the absolute difference image, $|im - bg|$, choosing the max value across all color channels, and normalizing the image into the range $[0; 255]$. The resulting image is filtered with a 5×5 median filter to reduce noise.

Top-View Detection

The top-view is thresholded based on the assumption of the image histogram being bimodal, due to a near uniform bright background, and darker zebrafish. Therefore the intermodal approach of Prewitt et al. [68] can be utilized. The threshold is set to the middle point between the two modes in the image histogram. If the histogram is not bimodal, the frame is filtered with a 3×3 mean filter, until its histogram is bimodal.

The fish are detected by applying a skeletonization based approach. BLOBs are initially detected and filled, and the skeletonization approach of Zhang and Suen [69] is applied. The skeleton is analyzed in order to find the skeleton keypoints: head, tail, and junctions. This is done by convolving the frame with the kernel in Equation (8). All end points of the skeleton have a value of 116, 117, 118, or 131, while junctions have a value of 148, 149, 150, or 151. Values, like 132, that can represent both an end point or an arbitrary point on the skeleton are not considered.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 15 & 15 & 15 & 1 \\ 1 & 15 & 100 & 15 & 1 \\ 1 & 15 & 15 & 15 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{8}$$

Each keypoint is assigned a weight, w, consisting of the smallest eigenvalue of the covariance matrix of the BLOB coordinates extracted from a 20×20 window around the keypoint. The smallest eigenvalue is used, as it is assumed the variance will be smallest along the width of the head. This way the head keypoints will be assigned a larger w than the tail keypoints, as the width of the zebrafish head is usually larger than the width of the zebrafish tail. The w of the junction points are reduced by a factor of 2.5, as they are usually larger than endpoint keypoints.

Keypoints too close to each other are removed by applying non-maximal suppression (NMS). Each keypoint is assigned a bounding box of size $w \times w$, and kept if the bounding box does not overlap with any other keypoint by

Parameter	c	δ_T	δ_F	τ_k	α	τ_p	β
Value	95	15	0.5/15	10	10	25	0.02

Table 3: Pipeline parameters used for testing.

more than $\text{NMS}_{\text{thresh}}$% of the area of the keypoint in focus. In case there is an overlap, only the keypoint with the largest w is kept. $\text{NMS}_{\text{thresh}}$ was set to 50 in the conducted tests.

Finally, all keypoints with $w < 1$ are discarded. Per skeleton, the two keypoints furthest apart are determined, and the keypoint with largest w is kept. Furthermore, half of the additional extra keypoints with the largest w values are also kept in order to handle crossings and occlusions. Every keypoint ideally represent a zebrafish head and they are all kept as individual detections.

Front-View Detection

The front frame cannot be thresholded based on the assumption of a bimodal distribution of the image histogram, as the stripes of the zebrafish results in a non-uniform appearance. Instead the frame is thresholded by finding the point maximizing the total entropy of the image [76]. The entropy is modeled as *background* and *foreground* entropy for each non-zero bin.

For bin k the background entropy, b_k, is calculated by

$$b_k = |\sum_{i=0}^{k} \frac{h(i)}{h_c(i)} \log(\frac{h(i)}{h_c(i)})|, \tag{9}$$

where h is the normalized image histogram and h_c is the cumulative histogram of h. The foreground entropy, w_k, is calculated by

$$w_k = |\sum_{i=k+1}^{255} \frac{h(i)}{1 - h_c(i)} \log(\frac{h(i)}{1 - h_c(i)})|, \tag{10}$$

resulting in the two entropy sets $B = \{b_0, b_1, .., b_{255}\}$ and $W = \{w_0, w_1, .., w_{255}\}$ which are combined into

$$E = B + W, \tag{11}$$

and the threshold, t, is then finally determined by

$$t = \underset{x}{\text{argmax}}\, E(x). \tag{12}$$

The BLOBs in the thresholded image are found through simple Connected Component Analysis (CCA). Only the $2N_{\text{fish}}$ largest BLOBs are kept, as long

	Property	Mean	Std. Dev.	Median
Trn	Reproj. Error $[px]$	8.03	5.26	7.59
	Speed $[cm/s^2]$	2.13	2.32	1.54
Val	Reproj. Error $[px]$	6.22	4.50	5.43
	Speed $[cm/s^2]$	2.02	2.37	1.41
Tst	Reproj. Error $[px]$	4.36	3.27	3.69
	Speed $[cm/s^2]$	2.11	1.93	1.58

Table 4: Statistics of the reprojection error and speed of the zebrafish, for each split, based on the ground truth annotations.

as the area of the BLOBs are larger than a predefined threshold. This allows for potential reflections being detected alongside the true detection of the fish. Otherwise, if only the N_{fish} largest BLOBs are considered for further analysis, it is possible that reflections are considered at the expense of real fish.

As it is not known where the head of the fish is, two points are saved as proxies for the head, together with the center-of-mass. If the width of the BLOB bounding box is larger than the height, the proxy points are $(\min(x), \mu(y))$ and $(\max(x), \mu(y))$, where x and y are the coordinates of the BLOB pixels, and μ is the arithmetic mean function. Otherwise, the proxy points are $(\mu(x), \min(y))$ and $(\mu(x), \max(y))$. The center-of-mass is further used for constructing 2D tracklets, as it is the most stable of the three proxy points.

B.3 FRCNN-H Object Detection

Two Faster R-CNN networks were trained to detect the zebrafish heads in each view, respectively. The official PyTorch implementation of Faster R-CNN with a ResNet50 based Feature Pyramid Network [77] backbone was utilized in both cases. The model was pretrained on the COCO train2017 dataset. The network was fine-tuned for 30 epochs, using stochastic gradient descent with momentum. A learning rate of 0.005, momentum of 0.9, weight decay of 0.0005, and batch size of 8 was used. The learning rate was "warmed up" during the first epoch, linearly interpolating the learning rate from 0.001 to 0.005. Each model was trained on an RTX 2080TI.

B.4 Pipeline Parameters

The full system pipeline has a set of parameters, which needs to be manually set. The parameters were chosen based on empirical investigation on the training data and they are shown in Table 3. The front-view distance threshold, δ_F, has two different values, depending on the method applied. For the

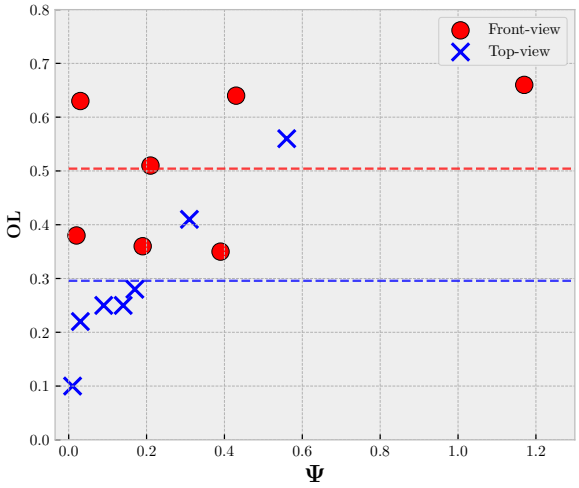


Fig. 9: Ψ is the proposed complexity measure based on the front- and top-view recordings. OL is the average length of occlusion events and the dashed lines show the mean OL for the two views. Tst1 is not included.

Naive method it is set to 0.5, as the distance is measured in standard deviations, whereas for the FRCNN-H method it is set to 15, where the distance is measured in pixels.

Furthermore, during the association of 2D tracklets between views, a set of exponential distributions are utilized. These distributions are parameterized using the mean of the reprojection error of the ground truth training annotations and the average speed of the zebrafish in the training split. This builds on the assumption that these parameters generalize across the different splits. As shown in Table 4, the average speed of the zebrafish for each split are within $0.1 \frac{cm}{s^2}$, which can be seen as a negligible difference, whereas the mean reprojection error is $2-4$ pixels larger in the training and validation splits than in the testing split. The cause of this difference is hypothesized to be the larger amount and duration of occlusions in the training and validation splits, which can introduce some uncertainty during the annotation phase. The training split has the largest reprojection error, which means the utilized exponential distribution penalize larger reprojection errors less harshly. However, as the mean training reprojection error is still low, it is still deemed fitting for the task.

B.5 Tracking Metrics and Results

When evaluating tracking tasks, a wide suite of metrics are often used, each expressing different properties of the task. Commonly, the MOTA metric is

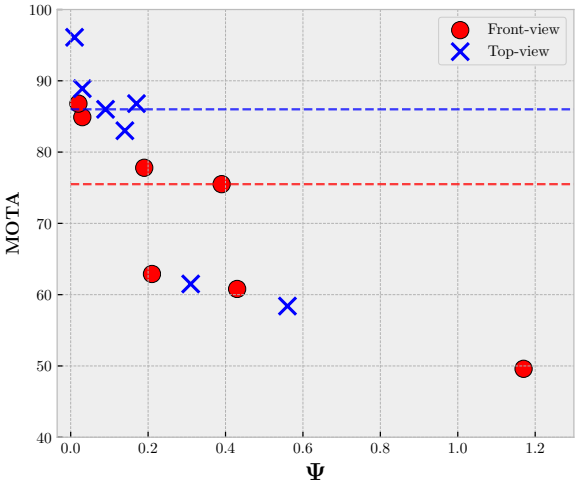


Fig. 10: Ψ is the proposed complexity measure based on the front- and top-view recordings. MOTA is based on the oracle tracker and the dashed lines show the mean MOTA for the two views. Tst1 is not included.

used as a representative metric for the overall performance of the tracker, as it incorporates false positives and negatives, as well as the amount of identity swaps. However, as shown by Carr and Collins [72] the MOTA metric is not guaranteed to be robust at all times. We evaluate the generated tracks using the collection of metrics from the MOT challenge, consisting of the CLEAR MOT [73], mostly tracked/lost metric [74], and identity based metrics of Ristani et al. [75], as well as using the MTBF metric proposed by Carr and Collins [72]. A description of each of the used metrics are presented in Table 5. Each step of the pipeline has been evaluated according to the previous mentioned metrics. Specifically the following steps have been evaluated:

- 2D tracklets after 2D Tracklet Construction (for top-view see Table 6 and for front-view see Table 7)
- 2D tracklets after 2D Tracklet Association Between Views (for top-view see Table 8 and for front-view see Table 9)
- 3D tracklets after 2D Tracklet Association Between Views (see Table 10)
- 3D tracks after 3D Tracklet Association (see Table 11)

For the 2D tracklets, a distance threshold of 20 px is used.

Occlusions

The two views are not equal when it comes to occlusions. Due to the social behavior of zebrafish they tend to swim in the same horizontal layer of the water column. This is indicated by the graph in Figure 9, where the occlusion lengths are displayed with respect to the proposed complexity measure, Ψ. It should be noted that Ψ is calculated per view and not as a mean between the two, as in the main article. Data points are presented for all the recordings from both views, except Tst1 as it only contains a single fish and therefore has a complexity measure of zero. The dashed lines illustrate the mean occlusion lengths and shows that there is a significant difference between the two views. In general, the occlusion events seems longer in the front-view recordings.

The MOTA of the hypothetical oracle tracker is plotted against the proposed complexity measure in Figure 10. MOTA is calculated for the top- and front-view 2D tracklets as presented in Table 6 and Table 7, respectively. The tracking performance of the oracle tracker correlates well with the complexity in both views. Notice the significant difference in performance of the tracker in the two views; the mean scores are $\text{MOTA}_{\text{top}} = 86.0\%$ and $\text{MOTA}_{\text{front}} = 75.5\%$ as illustrated by the two dashed lines.

Metric	Better	Range	Description
MOTA	Higher	$]-\infty, 100]$	Overall tracking accuracy based on ID-swaps, and false negatives/positives [73]
MOTP	Lower	$[0, \infty[$	Distance between predicted position and ground truth [73]
Prc.	Higher	$[0, 100]$	Precision: The percentage of correctly predicted positions
Rcll.	Higher	$[0, 100]$	Recall: The percentage of ground truth positions detected
ID-Prc.	Higher	$[0, 100]$	ID-Precision: The percentage of correctly identified predicted positions [75]
ID-Rcll.	Higher	$[0, 100]$	ID-Recall: The percentage of correctly identified ground truth positions [75]
ID-F1	Higher	$[0, 100]$	F1-score of the predicted identities [75]
FP	Lower	$[0, \infty[$	Amount of incorrectly predicted positions
FN	Lower	$[0, \infty[$	Amount of ground truth positions missed
MT	Higher	$[0, 100]$	Amount of ground truth tracks which are 80% or more correctly tracked [74]
ML	Lower	$[0, 100]$	Amount of ground truth tracks which are 20% or less correctly tracked [74]
ID Sw.	Lower	$[0, \infty[$	Total amount of identity switches [78]
Frag.	Lower	$[0, \infty[$	Total amount of track fragmentations
MTBF_s	Higher	$[0, \infty[$	Mean time between ID-switches or false negatives/positives [72]
MTBF_m	Higher	$[0, \infty[$	Monotonic version of MTBF_s [72]

Table 5: Description of the full collection of utilized tracking metrics.

	Method	MOTA ↑	MOTP ↓	Prc. ↑	Rcll. ↑	ID-Rcll. ↑	ID-Prc. ↑	ID-F1 ↑	FP ↓	FN ↓	MT ↑	ML ↓	ID Sw. ↓	Frag. ↓	MTBF_s ↑	MTBF_m ↑
Trn2	Naive	21.9%	3.309	57.2	91.0	10.7	6.7	8.2	9798	1301	2	0	136	428	26.316	14.109
	FRCNN-H	91.3%	2.128	98.8	92.8	12.4	13.2	12.8	156	1030	2	0	72	140	87.829	45.408
	Oracle	61.5%	0.000	100.0	63.0	6.7	10.6	8.2	0	5314	0	0	216	216	41.587	20.699
Trn5	Naive	38.2%	3.196	64.1	90.1	31.5	22.4	26.2	2274	447	5	0	58	124	27.020	14.792
	FRCNN-H	90.2%	2.038	99.1	91.6	31.9	34.5	33.2	39	378	5	0	24	50	73.607	38.523
	Oracle	58.4%	0.000	100.0	59.6	19.1	32.0	23.9	0	1819	0	0	52	52	47.035	23.726
Val2	Naive	59.8%	3.050	72.6	96.7	57.1	42.8	48.9	1315	118	2	0	13	44	68.275	36.653
	FRCNN-H	96.8%	2.165	99.2	97.9	75.9	76.8	76.3	30	76	2	0	10	17	167.810	92.737
	Oracle	88.9%	0.000	100.0	89.7	25.8	28.8	27.2	0	372	2	0	28	28	107.600	55.655
Val5	Naive	89.4%	3.552	93.1	97.0	51.7	49.6	50.6	330	135	5	0	19	36	92.083	52.619
	FRCNN-H	96.4%	2.782	99.5	97.2	48.3	49.4	48.8	22	128	5	0	15	23	147.567	83.528
	Oracle	86.0%	0.000	100.0	86.9	37.2	42.8	39.8	0	598	4	0	40	40	87.933	46.553
Tst1	Naive	83.1%	4.095	85.6	100.0	100.0	85.6	92.2	152	0	1	0	0	0	900.000	900.000
	FRCNN-H	85.1%	2.513	100.0	85.7	49.7	58.0	53.5	0	129	1	0	5	34	22.029	11.174
	Oracle	100.0%	0.000	100.0	100.0	100.0	100.0	100.0	0	0	1	0	0	0	900.000	900.000
Tst2	Naive	98.6%	3.771	99.6	99.1	99.1	99.6	99.3	8	17	2	0	0	10	148.583	81.045
	FRCNN-H	72.6%	2.443	100.0	74.1	39.0	52.7	44.8	0	467	1	0	26	93	14.032	7.053
	Oracle	96.1%	0.000	100.0	96.7	32.6	33.7	33.1	0	60	2	0	10	10	145.000	79.091
Tst5	Naive	82.8%	3.801	89.3	94.7	49.7	46.8	48.2	512	237	5	0	24	82	45.839	24.222
	FRCNN-H	70.5%	2.685	99.9	71.8	25.6	35.6	29.7	4	1271	0	0	52	145	21.384	10.836
	Oracle	83.0%	0.000	100.0	84.0	44.7	53.2	48.6	0	721	3	0	44	44	77.122	39.779
Tst10	Naive	83.4%	3.752	89.3	95.4	59.0	55.2	57.0	1026	418	10	0	49	138	52.329	28.323
	FRCNN-H	70.8%	2.803	99.8	72.4	26.4	36.4	30.6	13	2483	4	0	130	417	14.879	7.543
	Oracle	86.8%	0.000	100.0	87.5	42.1	48.1	44.9	0	1128	10	0	64	64	106.378	56.229

Table 6: Full metric evaluation of the 2D tracking in the top-view, on all sequences.

	Method	MOTA ↑	MOTP ↓	Prc. ↑	Rcll. ↑	ID-Rcll. ↑	ID-Prc. ↑	ID-F1 ↑	FP ↓	FN ↓	MT ↑	ML ↓	ID Sw. ↓	Frag. ↓	MTBF_s ↑	MTBF_m ↑
Trn2	Naive	-69.5%	12.666	7.6	6.2	0.6	0.7	0.6	10756	13490	0	2	134	159	4.709	2.528
	FRCNN-H	92.8%	3.521	99.8	95.0	6.4	6.8	6.6	26	723	2	0	288	126	43.218	30.898
	Oracle	62.9%	0.000	100.0	64.1	2.4	3.8	3.0	0	5168	0	0	170	170	53.558	26.936
Trn5	Naive	-73.5%	12.768	5.2	4.2	3.1	3.8	3.4	3478	4310	0	5	19	22	5.758	2.923
	FRCNN-H	90.7%	3.634	99.9	92.4	23.2	25.1	24.1	6	341	5	0	72	47	45.207	29.496
	Oracle	60.8%	0.000	100.0	61.7	20.1	32.6	24.9	0	1724	1	0	41	41	60.348	29.849
Val2	Naive	-93.1%	12.918	3.7	3.6	0.8	0.8	0.8	3444	3469	0	2	37	28	3.119	1.795
	FRCNN-H	93.6%	4.113	100.0	97.4	12.9	13.3	13.1	0	95	2	0	136	24	25.036	21.372
	Oracle	84.9%	0.000	100.0	85.3	16.3	19.1	17.6	0	530	2	0	14	14	191.875	102.333
Val5	Naive	-68.8%	12.419	10.0	8.5	2.9	3.4	3.1	3480	4162	0	5	37	44	6.690	3.464
	FRCNN-H	79.0%	4.224	99.6	81.8	16.2	19.7	17.8	14	826	3	0	117	72	26.791	17.402
	Oracle	49.6%	0.000	100.0	50.8	16.5	32.4	21.8	0	2240	0	0	54	54	39.153	19.914
Tst1	Naive	-44.8%	14.682	27.5	26.2	5.2	5.5	5.4	621	664	0	0	18	28	6.378	3.522
	FRCNN-H	41.4%	7.031	99.0	44.4	15.0	33.4	20.7	4	500	0	0	23	17	13.333	8.163
	Oracle	100.0%	0.000	100.0	100.0	100.0	100.0	100.0	0	0	1	0	0	0	900.000	900.000
Tst2	Naive	-38.3%	13.201	30.0	27.0	5.1	5.6	5.3	1135	1314	0	1	41	41	7.967	4.673
	FRCNN-H	16.4%	7.828	100.0	18.7	6.6	35.4	11.1	0	1464	0	1	40	57	5.695	2.824
	Oracle	86.8%	0.000	100.0	87.3	45.8	52.4	48.9	0	228	2	0	10	10	131.000	71.455
Tst5	Naive	-39.2%	12.703	28.5	24.3	4.6	5.4	4.9	2734	3408	0	3	124	138	5.717	3.250
	FRCNN-H	53.5%	5.601	98.7	56.4	15.4	27.0	19.7	33	1962	0	0	99	147	15.962	8.161
	Oracle	77.8%	0.000	100.0	78.7	32.6	41.4	36.5	0	957	2	0	42	42	75.383	38.934
Tst10	Naive	-54.1%	13.667	19.6	16.9	3.8	4.4	4.1	6219	7482	0	5	171	195	5.816	3.209
	FRCNN-H	47.4%	5.604	99.4	50.3	13.0	25.6	17.2	27	4474	0	0	229	277	14.368	7.518
	Oracle	75.5%	0.000	100.0	76.5	25.7	33.6	29.2	0	2115	4	0	94	94	66.202	33.261

Table 7: Full metric evaluation of the 2D tracking in the front-view, on all sequences.

	Method	MOTA ↑	MOTP ↓	Prc. ↑	Rcll. ↑	ID-Rcll. ↑	ID-Prc. ↑	ID-F1 ↑	FP ↓	FN ↓	MT ↑	ML ↓	ID Sw. ↓	Frag. ↓	$MTBF_s$ ↑	$MTBF_m$ ↑
Trn2	Naive	82.6%	3.247	96.2	86.5	13.2	14.6	13.9	497	1939	2	0	73	403	28.732	14.846
	FRCNN-H	90.9%	2.113	100.0	91.1	22.1	24.2	23.1	1	1273	2	0	41	144	85.667	43.836
	Oracle	44.8%	0.000	100.0	46.2	2.4	5.3	3.3	0	7742	0	0	202	202	32.539	16.190
Trn5	Naive	79.3%	3.151	95.9	83.4	37.5	43.2	40.1	159	748	3	0	23	119	29.543	15.129
	FRCNN-H	90.2%	2.028	100.0	90.5	45.6	50.4	47.8	0	427	5	0	13	44	81.460	42.874
	Oracle	36.6%	0.000	100.0	37.8	10.9	28.9	15.9	0	2799	0	0	53	53	29.328	14.175
Val2	Naive	89.0%	3.047	99.4	89.8	63.3	70.1	66.6	20	369	2	0	8	52	57.696	29.642
	FRCNN-H	96.1%	2.163	100.0	96.3	78.4	81.4	79.8	0	133	2	0	7	20	157.591	82.548
	Oracle	73.9%	0.000	100.0	74.9	12.7	17.0	14.5	0	902	0	0	36	36	71.000	35.500
Val5	Naive	92.1%	3.524	99.8	92.5	53.3	57.4	55.3	9	341	4	0	9	57	67.968	34.826
	FRCNN-H	90.9%	2.765	100.0	91.1	51.8	56.8	54.2	0	404	4	0	10	24	143.138	76.870
	Oracle	42.2%	0.000	100.0	43.6	13.2	30.3	18.4	0	2567	0	0	64	64	28.812	14.406
Tst1	Naive	99.6%	4.102	100.0	99.6	99.6	100.0	99.8	0	4	1	0	0	2	298.667	179.200
	FRCNN-H	69.8%	2.477	100.0	69.8	69.8	100.0	82.2	0	272	0	0	0	31	19.625	9.812
	Oracle	100.0%	0.000	100.0	100.0	100.0	100.0	100.0	0	0	1	0	0	0	900.000	900.000
Tst2	Naive	98.1%	3.782	99.9	98.1	98.1	99.9	99.0	1	34	2	0	0	18	88.300	46.474
	FRCNN-H	40.8%	2.514	100.0	40.9	32.8	80.1	46.5	0	1063	0	1	2	35	19.919	9.827
	Oracle	82.2%	0.000	100.0	83.2	24.3	29.2	26.6	0	302	2	0	18	18	74.900	37.450
Tst5	Naive	86.1%	3.737	98.3	87.9	52.8	59.1	55.8	67	546	4	0	13	69	51.351	26.716
	FRCNN-H	65.8%	2.666	100.0	66.2	34.0	51.3	40.9	0	1519	0	0	21	110	25.698	13.075
	Oracle	65.7%	0.000	100.0	66.8	26.8	40.1	32.2	0	1492	1	0	50	50	54.691	26.857
Tst10	Naive	87.0%	3.757	98.4	88.6	68.1	75.7	71.7	126	1023	9	0	23	151	48.054	24.697
	FRCNN-H	62.4%	2.797	100.0	62.9	31.4	49.9	38.6	2	3341	2	0	39	293	18.677	9.323
	Oracle	64.7%	0.000	100.0	66.0	20.0	30.3	24.1	0	3059	1	0	119	119	46.054	22.676

Table 8: Full metric evaluation of the 2D tracking in the top-view, after 3D tracklet association, on all sequences.

	Method	MOTA ↑	MOTP ↓	Prc. ↑	Rcll. ↑	ID-Rcll. ↑	ID-Prc. ↑	ID-F1 ↑	FP ↓	FN ↓	MT ↑	ML ↓	ID Sw. ↓	Frag. ↓	$MTBF_s$ ↑	$MTBF_m$ ↑
Trn2	Naive	-5.5%	14.688	46.6	33.4	6.6	9.2	7.7	5513	9577	0	0	82	772	6.174	3.091
	FRCNN-H	85.3%	3.482	99.9	85.6	19.5	22.8	21.0	10	2074	2	0	37	141	82.040	42.289
	Oracle	44.8%	0.000	100.0	46.2	2.4	5.3	3.3	0	7742	0	0	202	202	32.539	16.190
Trn5	Naive	2.2%	13.648	51.9	37.8	19.1	26.2	22.1	1577	2800	0	2	22	168	9.770	4.843
	FRCNN-H	81.5%	3.540	99.9	82.0	41.4	50.5	45.5	3	812	3	0	16	47	68.296	35.124
	Oracle	36.6%	0.000	100.0	37.8	10.9	28.9	15.9	0	2799	0	0	53	53	29.328	14.058
Val2	Naive	-7.9%	14.061	45.5	39.0	30.7	35.8	33.0	1678	2197	0	0	11	165	8.401	4.201
	FRCNN-H	82.1%	4.095	100.0	82.3	63.6	77.2	69.7	0	638	1	0	7	36	75.949	39.493
	Oracle	73.9%	0.000	100.0	74.9	12.7	17.0	14.5	0	902	0	0	36	36	71.000	35.500
Val5	Naive	-6.9%	11.925	43.9	24.0	13.9	25.5	18.0	1391	3460	0	2	15	108	9.646	4.781
	FRCNN-H	69.3%	4.068	99.9	69.7	35.0	50.1	41.2	4	1379	0	0	15	55	51.984	26.207
	Oracle	42.3%	0.000	100.0	43.7	13.3	30.3	18.4	0	2562	0	0	64	64	28.812	14.099
Tst1	Naive	72.1%	8.828	92.8	78.2	78.2	92.8	84.9	55	196	0	0	0	35	19.556	9.778
	FRCNN-H	37.8%	6.992	100.0	37.8	37.8	100.0	54.8	0	560	0	0	0	7	42.500	20.000
	Oracle	100.0%	0.000	100.0	100.0	100.0	100.0	100.0	0	0	1	0	0	0	900.000	900.000
Tst2	Naive	75.7%	6.560	95.6	79.4	79.4	95.6	86.7	66	371	1	0	0	38	35.725	18.089
	FRCNN-H	7.8%	6.187	100.0	7.9	4.8	61.3	9.0	0	1658	0	2	2	10	11.833	5.462
	Oracle	82.2%	0.000	100.0	83.2	24.3	29.2	26.6	0	302	2	0	18	18	74.900	37.450
Tst5	Naive	39.7%	9.665	77.9	56.1	38.6	53.6	44.9	717	1977	0	0	21	229	10.736	5.403
	FRCNN-H	50.1%	5.460	99.4	50.9	27.2	53.2	36.0	13	2209	0	0	23	91	23.619	11.749
	Oracle	65.7%	0.000	100.0	66.8	26.8	40.1	32.2	0	1492	1	0	50	50	54.691	27.345
Tst10	Naive	42.1%	10.218	78.3	59.0	45.4	60.3	51.8	1467	3693	0	0	47	339	15.034	7.506
	FRCNN-H	39.8%	5.488	99.9	40.3	19.6	48.8	28.0	3	5377	0	0	37	120	27.656	13.620
	Oracle	64.7%	0.000	100.0	66.0	20.0	30.3	24.1	0	3059	1	0	119	119	46.054	22.589

Table 9: Full metric evaluation of the 2D tracking in the front-view, after 3D tracklet association, on all sequences.

	Method	MOTA ↑	MOTP ↓	Prc. ↑	Rcll. ↑	ID-Rcll. ↑	ID-Prc. ↑	ID-F1 ↑	FP ↓	FN ↓	MT ↑	ML ↓	ID Sw. ↓	Frag. ↓	$MTBF_s$ ↑	$MTBF_m$ ↑
Trn2	Naive	41.1%	0.198	85.0	50.5	9.1	15.3	11.4	1285	7118	0	0	60	586	12.344	6.172
	FRCNN-H	73.7%	0.066	96.9	76.3	18.8	23.8	21.0	352	3407	0	0	29	215	50.548	25.333
	Oracle	44.8%	0.000	100.0	46.2	2.4	5.3	3.4	0	7738	0	0	202	202	32.539	16.190
Trn5	Naive	40.3%	0.169	83.5	50.5	25.5	42.1	31.8	448	2218	0	0	12	134	16.309	8.039
	FRCNN-H	63.5%	0.064	92.1	69.7	37.1	49.1	42.3	269	1358	1	0	9	67	43.431	21.867
	Oracle	36.7%	0.000	100.0	37.9	11.0	28.9	15.9	0	2784	0	0	53	53	29.328	14.175
Val2	Naive	59.7%	0.163	92.2	65.4	48.5	68.3	56.7	200	1244	0	0	5	86	26.682	13.341
	FRCNN-H	71.4%	0.062	96.3	74.4	61.4	79.5	69.3	103	919	0	0	4	36	70.342	36.122
	Oracle	74.1%	0.000	100.0	75.1	12.8	17.0	14.6	0	894	0	0	36	36	71.000	36.459
Val5	Naive	28.4%	0.204	78.8	39.1	23.2	46.8	31.1	477	2764	0	0	8	93	18.122	9.015
	FRCNN-H	61.3%	0.070	95.4	64.6	33.7	49.7	40.2	143	1607	1	0	9	60	45.123	22.562
	Oracle	42.4%	0.000	100.0	43.8	13.3	30.3	18.5	0	2552	0	0	64	64	28.812	14.618
Tst1	Naive	77.6%	0.152	96.0	80.9	80.9	96.0	87.8	30	171	1	0	0	28	25.000	12.500
	FRCNN-H	30.2%	0.105	100.0	30.2	30.2	100.0	46.4	0	625	0	0	0	15	16.938	8.212
	Oracle	100.0%	0.000	100.0	100.0	100.0	100.0	100.0	0	0	1	0	0	0	900.000	900.000
Tst2	Naive	77.6%	0.138	96.9	80.2	80.2	96.9	87.7	46	353	1	0	0	44	31.022	15.856
	FRCNN-H	5.7%	0.133	100.0	5.8	3.5	60.2	6.6	0	1677	0	2	2	17	5.421	2.641
	Oracle	80.6%	0.000	100.0	81.6	23.9	29.3	26.3	0	328	2	0	18	25	53.778	27.396
Tst5	Naive	39.7%	0.168	80.0	53.1	38.0	57.3	45.7	589	2078	0	0	9	186	12.340	6.219
	FRCNN-H	40.0%	0.099	98.0	41.3	23.2	55.1	32.6	37	2605	0	0	17	117	15.000	7.469
	Oracle	66.7%	0.000	100.0	67.8	27.2	40.1	32.4	0	1427	1	0	50	50	54.691	28.112
Tst10	Naive	48.2%	0.153	86.7	57.1	46.3	70.3	55.8	780	3824	0	0	16	273	18.007	8.925
	FRCNN-H	25.2%	0.099	97.0	26.3	14.1	52.0	22.2	72	6571	0	3	32	225	9.996	4.904
	Oracle	65.2%	0.000	100.0	66.6	20.2	30.3	24.3	0	2982	1	0	119	119	46.031	23.105

Table 10: Full metric evaluation of the 3D tracklets generated from the 3D tracklet association, on all sequences.

	Method	MOTA ↑	MOTP ↓	Prc. ↑	Rcll. ↑	ID-Rcll. ↑	ID-Prc. ↑	ID-F1 ↑	FP ↓	FN ↓	MT ↑	ML ↓	ID Sw. ↓	Frag. ↓	$MTBF_s$ ↑	$MTBF_m$ ↑
Trn2	Naive	41.4%	0.198	85.3	50.4	28.5	48.3	35.9	1250	7133	0	0	40	573	12.597	6.298
	FRCNN-H	73.8%	0.066	96.9	76.3	44.0	55.9	49.2	352	3407	0	0	14	216	50.317	25.216
	Oracle	46.2%	0.000	100.0	46.2	46.2	100.0	63.2	0	7738	0	0	0	202	32.539	16.190
Trn5	Naive	40.4%	0.170	83.8	50.3	32.5	54.2	40.7	436	2228	0	0	7	135	16.121	7.947
	FRCNN-H	63.5%	0.064	92.1	69.7	40.9	54.1	46.6	269	1359	1	0	7	66	44.028	22.170
	Oracle	37.9%	0.000	100.0	37.9	37.9	100.0	55.0	0	2784	0	0	0	53	29.328	14.175
Val2	Naive	59.6%	0.163	92.2	65.3	50.8	71.7	59.4	199	1248	0	0	3	82	27.905	13.870
	FRCNN-H	71.5%	0.062	96.3	74.4	71.7	92.8	80.9	103	920	0	0	2	35	72.216	37.111
	Oracle	75.1%	0.000	100.0	75.1	75.1	100.0	85.8	0	894	0	0	0	36	71.000	36.459
Val5	Naive	28.5%	0.204	78.9	39.1	23.2	46.8	31.1	476	2764	0	0	7	93	18.122	9.015
	FRCNN-H	61.3%	0.070	95.4	64.6	40.2	59.4	48.0	143	1607	1	0	5	60	45.123	22.562
	Oracle	43.8%	0.000	100.0	43.8	43.8	100.0	60.9	0	2552	0	0	0	64	28.812	14.618
Tst1	Naive	77.6%	0.152	96.0	80.9	80.9	96.0	87.8	30	171	1	0	0	28	25.000	12.500
	FRCNN-H	30.2%	0.105	100.0	30.2	30.2	100.0	46.4	0	625	0	0	0	15	16.938	8.212
	Oracle	100.0%	0.000	100.0	100.0	100.0	100.0	100.0	0	0	1	0	0	0	900.000	900.000
Tst2	Naive	77.6%	0.138	96.9	80.2	80.2	96.9	87.7	46	353	1	0	0	44	31.022	15.856
	FRCNN-H*	5.7%	0.133	100.0	5.8	3.5	60.2	6.6	0	1677	0	2	2	17	5.421	2.641
	Oracle	81.6%	0.000	100.0	81.6	81.6	100.0	89.9	0	328	2	0	0	25	53.778	27.396
Tst5	Naive	39.7%	0.168	80.0	53.1	43.4	65.4	52.2	588	2079	0	0	7	185	12.400	6.249
	FRCNN-H	40.2%	0.099	98.0	41.2	29.8	70.9	41.9	37	2609	0	0	7	115	15.217	7.577
	Oracle	67.8%	0.000	100.0	67.8	67.8	100.0	80.8	0	1427	1	0	0	50	54.691	28.112
Tst10	Naive	48.3%	0.153	86.9	57.1	48.5	73.8	58.5	768	3829	0	0	11	268	18.313	9.075
	FRCNN-H*	25.2%	0.099	97.0	26.3	14.1	52.0	22.2	72	6571	0	3	32	225	9.996	4.904
	Oracle	66.6%	0.000	100.0	66.6	66.6	100.0	79.9	0	2982	1	0	0	119	46.031	23.105

Table 11: Full metric evaluation of the final 3D tracks generated from the 3D track association, on all sequences. An * indicates that the method did not complete the 3D Tracklet Association step for the sequence, as there were not enough concurrent 3D tracklets at any one point in the sequence. In this case, the 3D tracklet results are reported.

References

[1] J. S. Eisen, "Zebrafish make a big splash," *Cell*, vol. 87, no. 6, pp. 969–977, Dec. 1996.

[2] P. Haffter, M. Granato, M. Brand, M. Mullins, M. Hammerschmidt, D. Kane, J. Odenthal, F. van Eeden, Y. Jiang, C. Heisenberg, R. Kelsh, M. Furutani-Seiki, E. Vogelsang, D. Beuchle, U. Schach, C. Fabian, and C. Nusslein-Volhard, "The identification of genes with unique and essential functions in the development of the zebrafish, Danio rerio," *Development*, vol. 123, no. 1, pp. 1–36, 1996. [Online]. Available: https://dev.biologists.org/content/123/1/1

[3] A. D. Collier, K. M. Khan, E. M. Caramillo, R. S. Mohn, and D. J. Echevarria, "Zebrafish and conditioned place preference: A translational model of drug reward," *Progress in Neuro-Psychopharmacology and Biological Psychiatry*, vol. 55, pp. 16–25, Dec. 2014.

[4] C.-Y. Lin, C.-Y. Chiang, and H.-J. Tsai, "Zebrafish and Medaka: new model organisms for modern biomedical research," *Journal of Biomedical Science*, vol. 23, no. 1, Jan. 2016.

[5] M. C. Soares, S. C. Cardoso, T. d. S. Carvalho, and C. Maximino, "Using model fish to study the biological mechanisms of cooperative behaviour: A future for translational research concerning social anxiety disorders?" *Progress in Neuro-Psychopharmacology and Biological Psychiatry*, vol. 82, pp. 205–215, Mar. 2018.

[6] A. V. Kalueff, A. M. Stewart, and R. Gerlai, "Zebrafish as an emerging model for studying complex brain disorders," *Trends in Pharmacological Sciences*, vol. 35, no. 2, pp. 63–75, Feb. 2014.

[7] D. A. Meshalkina, M. N. Kizlyk, E. V. Kysil, A. D. Collier, D. J. Echevarria, M. S. Abreu, L. J. G. Barcellos, C. Song, J. E. Warnick, E. J. Kyzar, and A. V. Kalueff, "Zebrafish models of autism spectrum disorder," *Experimental Neurology*, vol. 299, pp. 207–216, Jan. 2018.

[8] K. M. Khan, A. D. Collier, D. A. Meshalkina, E. V. Kysil, S. L. Khatsko, T. Kolesnikova, Y. Y. Morzherin, J. E. Warnick, A. V. Kalueff, and D. J. Echevarria, "Zebrafish models in neuropsychopharmacology and CNS drug discovery," *British Journal of Pharmacology*, vol. 174, no. 13, pp. 1925–1944, 2017.

[9] L. Li and J. E. Dowling, "A dominant form of inherited retinal degeneration caused by a non-photoreceptor cell-specific mutation," *Proceedings of the National Academy of Sciences*, vol. 94, no. 21, pp. 11 645–11 650, Oct. 1997.

[10] U. K. Muller, "Swimming of larval zebrafish: ontogeny of body waves and implications for locomotory development," *Journal of Experimental Biology*, vol. 207, no. 5, pp. 853–868, Feb. 2004.

[11] M. B. McElligott and D. M. O'Malley, "Prey tracking by larval zebrafish: Axial kinematics and visual control," *Brain, Behavior and Evolution*, vol. 66, no. 3, pp. 177–196, 2005.

[12] E. Fontaine, D. Lentink, S. Kranenbarg, U. K. Müller, J. L. van Leeuwen, A. H. Barr, and J. W. Burdick, "Automated visual tracking for studying the ontogeny of

zebrafish swimming," *Journal of Experimental Biology*, vol. 211, no. 8, pp. 1305–1316, 2008.

[13] B. Risse, D. Berh, N. Otto, C. Klämbt, and X. Jiang, "FIMTrack: An open source tracking and locomotion analysis software for small animals," *PLOS Computational Biology*, vol. 13, no. 5, pp. 1–15, 2017.

[14] S. Ohayon, O. Avni, A. L. Taylor, P. Perona, and S. E. R. Egnor, "Automated multi-day tracking of marked mice for the analysis of social behaviour," *Journal of Neuroscience Methods*, vol. 219, no. 1, pp. 10 – 19, 2013.

[15] Z.-M. Qian, X. E. Cheng, and Y. Q. Chen, "Automatically detect and track multiple fish swimming in shallow water with frequent occlusion," *PLOS ONE*, vol. 9, no. 9, pp. 1–12, 2014.

[16] G. d. O. Feijó, V. A. Sangalli, I. N. L. d. Silva, and M. S. Pinho, "An algorithm to track laboratory zebrafish shoals," *Computers in Biology and Medicine*, vol. 96, pp. 79 – 90, 2018.

[17] J. E. Franco-Restrepo, D. A. Forero, and R. A. Vargas, "A review of freely available, open-source software for the automated analysis of the behavior of adult zebrafish," *Zebrafish*, vol. 16, no. 3, Jun. 2019.

[18] F. Romero-Ferrero, M. G. Bergomi, R. C. Hinz, F. J. H. Heras, and G. G. d. Polavieja, "idtracker.ai: tracking all individuals in small or large collectives of unmarked animals," *Nature Methods*, vol. 16, no. 2, pp. 179–182, jan 2019.

[19] L. P. J. J. Noldus, A. J. Spink, and R. A. J. Tegelenbosch, "EthoVision: A versatile video tracking system for automation of behavioral experiments," *Behavior Research Methods, Instruments, & Computers*, vol. 33, no. 3, pp. 398–414, Aug. 2001.

[20] Loligo Systems, "LoliTrack v.4," https://www.loligosystems.com/lolitrack-v-4.

[21] ViewPoint, "ZebraLab," http://www.viewpoint.fr/en/p/software/zebralab.

[22] TSE Systems, "VideoMot2 - versatile video tracking system," https://www.tse-systems.com/product-details/videomot.

[23] A. V. Kalueff, M. Gebhardt, A. M. Stewart, J. M. Cachat, M. Brimmer, J. S. Chawla, C. Craddock, E. J. Kyzar, A. Roth, S. Landsman, S. Gaikwad, K. Robinson, E. Baatrup, K. Tierney, A. Shamchuk, W. Norton, N. Miller, T. Nicolson, O. Braubach, C. P. Gilman, J. Pittman, D. B. Rosemberg, R. Gerlai, D. Echevarria, E. Lamb, S. C. F. Neuhauss, W. Weng, L. Bally-Cuif, H. Schneider, and t. Z. Neuros, "Towards a comprehensive catalog of zebrafish behavior 1.0 and beyond," *Zebrafish*, vol. 10, no. 1, pp. 70–86, Mar. 2013.

[24] J. M. Cachat, P. R. Canavello, S. I. Elkhayat, B. K. Bartels, P. C. Hart, M. F. Elegante, E. C. Beeson, A. L. Laffoon, W. A. Haymore, D. H. Tien, A. K. Tien, S. Mohnot, and A. V. Kalueff, "Video-aided analysis of zebrafish locomotion and anxiety-related behavioral responses," in *Zebrafish Neurobehavioral Protocols*. Totowa, NJ: Humana Press, 2011, pp. 1–14.

[25] S. Macrí, D. Neri, T. Ruberto, V. Mwaffo, S. Butail, and M. Porfiri, "Three-dimensional scoring of zebrafish behavior unveils biological phenomena hidden by two-dimensional analyses," *Nature*, vol. 7, no. 1, may 2017.

[26] N. Miller and R. Gerlai, "Quantification of shoaling behaviour in zebrafish (Danio rerio)," *Behavioural Brain Research*, vol. 184, no. 2, pp. 157–166, Dec. 2007.

[27] H. AlZu'bi, W. Al-Nuaimy, J. Buckley, L. Sneddon, and Iain Young, "Real-time 3D fish tracking and behaviour analysis," in *Proceedings of the IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, Nov. 2015, pp. 1–5.

[28] X. E. Cheng, S. S. Du, H. Y. Li, J. F. Hu, and M. L. Chen, "Obtaining three-dimensional trajectory of multiple fish in water tank via video tracking," *Multimedia Tools and Applications*, vol. 77, no. 18, pp. 24 499–24 519, Feb. 2018.

[29] Z. Qian, M. Shi, M. Wang, and T. Cun, "Skeleton-based 3D tracking of multiple fish from two orthogonal views," in *Communications in Computer and Information Science*. Springer Singapore, 2017, pp. 25–36.

[30] S. H. Wang, X. Liu, J. Zhao, Y. Liu, and Y. Q. Chen, "3D tracking swimming fish school using a master view tracking first strategy," in *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, Dec. 2016.

[31] Z.-M. Qian and Y. Q. Chen, "Feature point based 3D tracking of multiple fish from multi-view images," *PLOS ONE*, vol. 12, no. 6, pp. 1–18, 2017.

[32] G. Audira, B. Sampurna, S. Juniardi, S.-T. Liang, Y.-H. Lai, and C.-D. Hsiao, "A simple setup to perform 3D locomotion tracking in zebrafish by using a single camera," *Inventions*, vol. 3, no. 1, p. 11, Feb. 2018.

[33] G. Xiao, W.-K. Fan, J.-F. Mao, Z.-B. Cheng, D.-H. Zhong, and Y. Li, "Research of the fish tracking method with occlusion based on monocular stereo vision," in *Proceedings of the International Conference on Information System and Artificial Intelligence (ISAI)*. IEEE, Jun. 2016.

[34] M. Menze, C. Heipke, and A. Geiger, "Joint 3D estimation of vehicles and scene flow," in *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.

[35] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "MOTS: multi-object tracking and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7934–7943.

[36] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: a multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.

[37] "Waymo open dataset: An autonomous driving dataset," 2019, https://www.waymo.com/open.

[38] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[39] S. Song and J. Xiao, "Tracking revisited using RGBD camera: Unified benchmark and baselines," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 233–240.

[40] A. Lukezic, U. Kart, J. Kapyla, A. Durmush, J.-K. Kamarainen, J. Matas, and M. Kristan, "Cdtb: A color and depth visual object tracking dataset and benchmark," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 10 013–10 022.

[41] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking," *arXiv:1504.01942*, apr 2015.

[42] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv:1603.00831*, mar 2016.

[43] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "CVPR19 tracking and detection challenge: How crowded can it get?" *arXiv:1906.04567 [cs]*, Jun. 2019.

[44] S. Lyu, M. Chang, D. Du, L. Wen, H. Qi, Y. Li, Y. Wei, L. Ke, T. Hu, M. Del Coco, P. Carcagnì, D. Anisimov, E. Bochinski, F. Galasso, F. Bunyak, G. Han, H. Ye, H. Wang, K. Palaniappan, K. Ozcan, L. Wang, L. Wang, M. Lauer, N. Watcharapinchai, N. Song, N. M. Al-Shakarji, S. Wang, S. Amin, S. Rujikietgumjorn, T. Khanova, T. Sikora, T. Kutschbach, V. Eiselein, W. Tian, X. Xue, X. Yu, Y. Lu, Y. Zheng, Y. Huang, and Y. Zhang, "UA-DETRAC 2017: Report of AVSS2017 IWT4S challenge on advanced traffic monitoring," in *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Aug 2017, pp. 1–7.

[45] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang, and S. Lyu, "Ua-detrac: A new benchmark and protocol for multi-object detection and tracking," *Computer Vision and Image Understanding*, vol. 193, p. 102907, 2020.

[46] S. Sun, N. Akhtar, H. Song, A. S. Mian, and M. Shah, "Deep affinity network for multiple object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, pp. 1–1, 2019.

[47] E. Bochinski, T. Senst, and T. Sikora, "Extending IOU based multi-object tracking by visual information," in *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Nov 2018, pp. 1–6.

[48] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2019, pp. 941–951.

[49] A. Pérez-Escudero, J. Vicente-Page, R. C. Hinz, S. Arganda, and G. G. De Polavieja, "idtracker: tracking individuals in a group by automatic identification of unmarked animals," *Nature methods*, vol. 11, no. 7, pp. 743–748, 2014.

[50] V. H. Sridhar, D. G. Roche, and S. Gingins, "Tracktor: Image-based automated tracking of animal movement and behaviour," *Methods in Ecology and Evolution*, vol. 10, no. 6, pp. 815–820, Mar. 2019.

[51] H. J. Mönck, A. Jörg, T. v. Falkenhausen, J. Tanke, B. Wild, D. Dormagen, J. Piotrowski, C. Winklmayr, D. Bierbach, and T. Landgraf, "BioTracker: an open-source computer vision framework for visual animal tracking," *arXiv:1803.07985*, 2018.

[52] A. Rodriguez, H. Zhang, J. Klaminder, T. Brodin, P. L. Andersson, and M. Andersson, "ToxTrac: a fast and robust software for tracking organisms," *Methods in Ecology and Evolution*, vol. 9, no. 3, pp. 460–464, Sep. 2017.

[53] A. M. T. Harmer and D. B. Thomas, "pathtrackr: An r package for video tracking and analysing animal movement," *Methods in Ecology and Evolution*, May 2019.

[54] X. Liu, P. R. Zhu, Y. Liu, and J. W. Zhao, "Tracking full-body motion of multiple fish with midline subspace constrained multicue optimization," *Scientific Programming*, vol. 2019, pp. 1–7, Jun. 2019.

[55] L. Zhu and W. Weng, "Catadioptric stereo-vision system for the real-time monitoring of 3D behavior in aquatic animals," *Physiology & Behavior*, vol. 91, no. 1, pp. 106 – 119, 2007.

[56] J. Cachat, A. Stewart, E. Utterback, P. Hart, S. Gaikwad, K. Wong, E. Kyzar, N. Wu, and A. V. Kalueff, "Three-dimensional neurophenotyping of adult zebrafish behavior," *PLOS ONE*, vol. 6, no. 3, pp. 1–14, 2011.

[57] X. E. Cheng, S. H. Wang, and Y. Q. Chen, "3D tracking targets via kinematic model weighted particle filter," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2016, pp. 1–6.

[58] S. V. Viscido, J. K. Parrish, and D. Grünbaum, "Individual behavior and emergent properties of fish schools: a comparison of observation and theory," *Marine Ecology Progress Series*, vol. 273, pp. 239–249, 2004.

[59] A. D. Straw, K. Branson, T. R. Neumann, and M. H. Dickinson, "Multi-camera real-time three-dimensional tracking of multiple flying animals," *Journal of The Royal Society Interface*, vol. 8, no. 56, pp. 395–409, Jul. 2010.

[60] K. Müller, J. Schlemper, L. Kuhnert, and K. D. Kuhnert, "Calibration and 3D ground truth data generation with orthogonal camera-setup and refraction compensation for aquaria in real-time," in *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, vol. 3, Jan. 2014, pp. 626–634.

[61] A. M. Stewart, F. Grieco, R. A. J. Tegelenbosch, E. J. Kyzar, M. Nguyen, A. Kaluyeva, C. Song, L. P. J. J. Noldus, and A. V. Kalueff, "A novel 3D method of locomotor analysis in adult zebrafish: Implications for automated detection of CNS drug-evoked phenotypes," *Journal of Neuroscience Methods*, vol. 255, pp. 66–74, Nov. 2015.

[62] Z.-M. Qian, S. H. Wang, X. E. Cheng, and Y. Q. Chen, "An effective and robust method for tracking multiple fish in video image based on fish head detection," *BMC Bioinformatics*, vol. 17, no. 1, p. 251, Jun. 2016.

[63] X. Liu, Y. Yue, M. Shi, and Z.-M. Qian, "3-D video tracking of multiple fish in a water tank," *IEEE Access*, vol. 7, pp. 145 049–145 059, 2019.

[64] S. H. Wang, X. E. Cheng, and Y. Q. Chen, "Tracking undulatory body motion of multiple fish based on midline dynamics modeling," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, July 2016, pp. 1–6.

[65] S. H. Wang, J. Zhao, X. Liu, Z. Qian, Y. Liu, and Y. Q. Chen, "3D tracking swimming fish school with learned kinematic model using LSTM network," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 1068–1072.

[66] S. H. Wang, J. W. Zhao, and Y. Q. Chen, "Robust tracking of fish schools using CNN for head identification," *Multimedia Tools and Applications*, pp. 1–19, 2016.

[67] M. Pedersen, S. Hein Bengtson, R. Gade, N. Madsen, and T. B. Moeslund, "Camera calibration for underwater 3D reconstruction based on ray tracing using Snell's law," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2018.

[68] J. M. S. Prewitt and M. L. Mendelsohn, "The analysis of cell images," *Annals of the New York Academy of Sciences*, vol. 128, no. 3, pp. 1035–1053, 1966.

[69] T. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.

[70] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28. Curran Associates, Inc., 2015, pp. 91–99.

[71] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[72] P. Carr and R. T. Collins, "Assessing tracking performance in complex scenarios using mean time between failures," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, March 2016, pp. 1–10.

[73] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, pp. 1–10, May 2008.

[74] B. Wu and R. Nevatia, "Tracking of multiple, partially occluded humans based on static body part detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2006, pp. 951–958.

[75] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Computer Vision – ECCV 2016 Workshops*. Springer, 2016, pp. 17–35.

[76] P. Sahoo, S. Soltani, and A. Wong, "A survey of thresholding techniques," *Computer Vision, Graphics, and Image Processing*, vol. 41, no. 2, pp. 233–260, Feb. 1988.

[77] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017.

[78] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: HybridBoosted multi-target tracker for crowded scene," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2009.

Paper C

MOTCOM: The Multi-Object Tracking Dataset Complexity Metric

Malte Pedersen, Joakim Bruslund Haurum,
Patrick Dendorfer, and Thomas B. Moeslund

The paper has been published in
Computer Vision – ECCV 2022, LNCS 13668, pp. 20-37.

The layout has been revised.

Abstract

There exists no comprehensive metric for describing the complexity of Multi-Object Tracking (MOT) sequences. This lack of metrics decreases explainability, complicates comparison of datasets, and reduces the conversation on tracker performance to a matter of leader board position. As a remedy, we present the novel MOT dataset complexity metric (MOTCOM), which is a combination of three sub-metrics inspired by key problems in MOT: occlusion, erratic motion, and visual similarity. The insights of MOTCOM can open nuanced discussions on tracker performance and may lead to a wider acknowledgement of novel contributions developed for either less known datasets or those aimed at solving sub-problems.

We evaluate MOTCOM on the comprehensive MOT17, MOT20, and MOTSynth datasets and show that MOTCOM is far better at describing the complexity of MOT sequences compared to the conventional density and number of tracks. Project page at `https://vap.aau.dk/motcom`.

1 Introduction

Tracking has been an important research topic for decades with applications ranging from autonomous driving to fish behavior analysis [1–4]. The aim is to acquire the full spatio-temporal trajectory of an object of interest, but missing or inaccurate detections can make this a complicated task. When more objects are present in the scene simultaneously it is termed a multi-object tracking (MOT) problem and an additional task is to keep the correct identities of all objects throughout the sequence.

During the previous decade there has been an increase in the development of publicly available MOT datasets [5–9]. However, there has been no attempt to objectively describe the complexity of a dataset or its sequences except for using simple statistics like *density* and *number of tracks*, which are neither adequate nor explanatory, see Figure C.1. When a new dataset emerges, the community needs objective metrics to be able to characterize and discuss the dataset with respect to existing datasets, otherwise, 'gut feeling' and 'popularity vote' will rule. Furthermore, the absence of an objective MOT sequence complexity metric hinders an informed conversation on the capabilities of trackers developed for different datasets. Nowadays, it is important to rank high on popular MOT benchmark leaderboads in order to gain the attention of the community. This may hinder the acknowledgement of novel solutions that solve sub-problems of MOT particularly well and underrate solutions developed on less popular datasets. We expect that a descriptive and explanatory metric can help remedy these issues.

The literature suggests that there are three main factors that make MOT tasks difficult to solve [2, 12–15]; namely, occlusion, erratic motion, and vi-

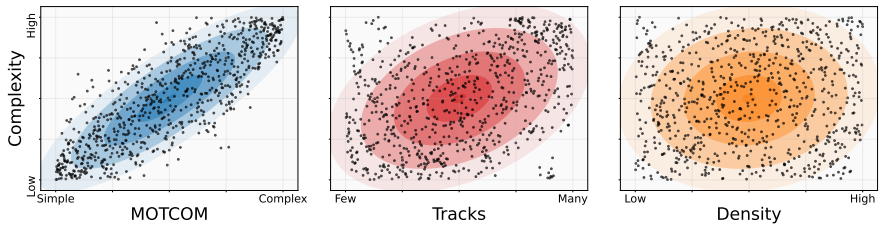


Fig. C.1: Comparing the capability of the proposed MOTCOM metric against the conventional metrics (*number of tracks* and *density*) for describing MOT sequence complexity. The shared y-axis shows a HOTA [10] rank-based proxy for the ground truth complexity of the MOTSynth sequences [11]. The x-axes show the corresponding rank determined by each of the three metrics. The correlation between the complexity and MOTCOM is clearly stronger compared to both *tracks* and *density*. More details can be found in Section 5.

sual similarity. We hypothesize that the complexity of MOT sequences can be expressed by a combination of the aforementioned three factors for which we need to construct explicit metrics. Therefore, in this paper we propose the first-ever individual sub-metrics for describing the complexity of the three sub-problems and a unified quantitative MOT dataset complexity metric (MOTCOM) as a combination of these sub-metrics. In Figure C.1, we illustrate that MOTCOM is far better at estimating the complexity of the sequences of the recent MOTSynth dataset [11] compared to the commonly used *number of tracks* and *density*.

The main contributions of our paper are as follows:

1. The novel metric MOTCOM for describing the complexity of MOT sequences.
2. Three sub-metrics for describing the complexity of MOT sequences with respect to occlusion, erratic motion, and visual similarity.
3. We show that the conventional metrics *number of tracks* and *density* are not strong indicators for the complexity of MOT sequences.
4. We evaluate the capability of MOTCOM and demonstrate its superiority against *number of tracks* and *density*.

In the next section, we describe and analyse the three sub-problems followed by a presentation of the proposed metrics. In the remainder of the paper, we demonstrate and discuss how the metrics can describe and explain the complexity of MOT sequences.

2 Related Work

The majority of recent trackers utilize the strong performance of deep learning based detectors, e.g., by following the tracking-by-detection paradigm [16–18], tracking-by-regression [12], through joint training of the detection and tracking steps [19, 20], or as part of an association step [21–23]. Trackers like Tracktor [12], Chained-Tracker [19], and CenterTrack [20] rely on spatial proximity which makes them vulnerable to sequences with extreme motion and heavy occlusion. At the other end of the spectrum are trackers like QDTrack [21], RetinaTrack [22], and FairMOT [23] which use visual cues for tracking. They are optimized toward tracking visually distinct objects and are not to the same degree limited by erratic motion or vanishing objects but instead sensitive to weak visual features. This indicates that the design of trackers is centered around three core problems: occlusion, erratic motion, and visual similarity. Below, we dive into the literature regarding these problems followed by insights on dataset complexity.

Occlusion.

Occlusions can be difficult to handle and they are often simply treated as missing data [15]. However, in scenes were the objects have weak or similar visual features this can be harmful for the tracking performance [14, 24, 25].

Most authors state that a higher occlusion rate makes tracking harder [26–28], but they seldom quantify such statements. An exception is the work proposed by Bergmann et al. [12] where they analyzed the tracking results with respect to object visibility, the size of the objects, and missing detections. Moreover, Pedersen et al. [13] argued that the number of objects is less critical than the amount and level of occlusion when it comes to multi-object tracking of fish. They described the complexity of their sequences based on occlusions alone.

Erratic Motion.

Prior information can be used to predict the next state of an object which minimizes the search space and hence reduces the impact of noisy or missing detections. A linear motion model assuming constant velocity is a simple, but effective method for predicting the movement of non-erratic objects like pedestrians [2, 24]. In scenes that include camera motion or complex movement more advanced models may improve tracker performance. Pellegrini et al. [29] proposed incorporating human social behavior into their motion model and Kratz et al. [30] proposed utilizing the movement of a crowd to enhance the tracking of individuals. A downside of many advanced motion models is an often poor ability to generalize to other types of objects or environments.

Visual Similarity.

Visual cues are commonly used in tracklet association and re-identification and are well studied for persons [31], vehicles [32], and animals [33] such as zebrafish [34] and tigers [35]. Modern trackers often solve the association step using CNNs, like Siamese networks, based on a visual affinity model [12, 36–38]. Such methods rely on visual dissimilarity between the objects. However, tracklet association becomes more difficult when objects are hard to distinguish purely by their appearance.

Dataset Complexity.

Determining the complexity of a dataset is a non-trivial task. One may have a "feeling" or intuition about which datasets are harder than others, but this is subjective and can differ depending on who you ask, as well as differ depending on the task at hand. In order to objectively determine the complexity of a dataset, one has to develop a task-specific framework. An early attempt at this was the suite of 12 complexity measures (c-measures) by Ho and Basu [39], based on concepts such as inter-class overlap and linear separability. However, these c-measures are not suitable for image datasets due to unrealistic assumptions, such as the data being linearly separable. Therefore, Branchaud-Charron et al. [40] developed a complexity measure based on spectral clustering, where the inter-class overlap is quantified through the eigenvalues of an approximated adjacency matrix. This approach was shown to correlate well with the CNN performance on several image datasets. Similarly, Cui et al. [41] presented a framework for evaluating the fine-grainedness of image datasets, by measuring the average distance from data examples to the class centers. Both of these approaches rely on embedding the input images into a feature space by using, e.g., a CNN, and determining the dataset complexity without any indication of what makes the dataset difficult.

In contrast, dataset complexity in the MOT field has so far been determined through simple statistics such as the number of tracks and density. These quantities are currently displayed for every sequence alongside other stats such as resolution and frame rate for the MOTChallenge benchmark datasets [7]. The preliminary works of Bergmann et al. [12] and Pedersen et al. [13] have attempted to further explain what makes a MOT sequence difficult by investigating the effect of occlusions. However, there is no clear way of describing the complexity of MOT sequences and the current methods have not been verified.

3 Challenges in Multi-Object Tracking

MOT covers the task of obtaining the spatio-temporal trajectories of multiple objects in a sequence of consecutive frames. Depending on the specific task, the objects may be represented as 3D points [13], pixel-level segmentation masks [42], or bounding boxes [43]. Despite the different representation forms, the concepts of occlusion, erratic motion, and visual similarity apply to all of them and add to the complexity of the sequences.

Occlusion.

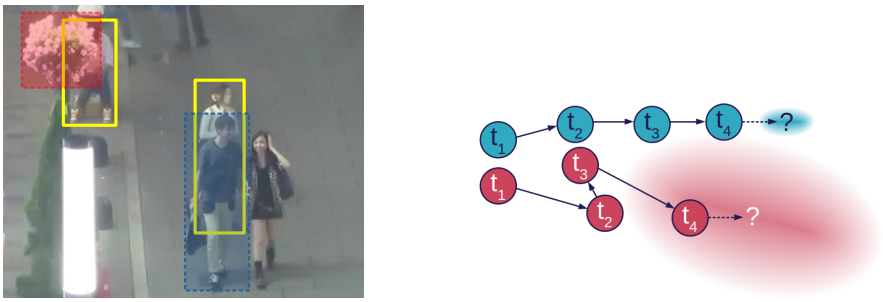
Occlusion describes situations where the visual information of an object within the camera view is partially or fully hidden. There are three types of occlusion: *self-occlusion*, *scene-occlusion*, and *inter-object-occlusion* [14]. Self-occlusion can reduce the visibility of parts of an object, e.g., if a hand is placed in front of a face, but defining the level of self-occlusion is non-trivial and depends on the type of object. Scene-occlusion occurs when a static object is located in the line of sight between the camera and the target object, thereby decreasing the visual information of the target. A scene-occlusion is marked by the red box in Figure C.2a, where flowers partially occlude a sitting person.

Erratic Motion.

We use motion as a term for an object's spatial displacement between frames. This is typically caused by the locomotive behavior of the object itself, camera motion, or a combination. As the number of factors that influence the observed motion increases, the motion becomes harder to predict. An example of two objects exhibiting different types of motion is presented in Figure C.2b. The blue object moves with approximately the same direction and speed between the time steps. Predicting the next state of the object seems trivial and the search space is correspondingly small. On the other hand, the red object behaves erratically and unpredictably while the motion model is less confident as illustrated by the larger search space.

Visual Similarity.

The visual appearance of objects can vary widely depending on the type of object and type of scene. Appearance is especially important when tracking is lost, for example, due to occlusion, and re-identification is a common tool for associating broken tracklets. The complexity of this process depends on the visual similarity between objects, but intra-object similarity also plays a role. As an object moves through a scene, its appearance can change from the perspective of the viewer. The object may turn around, increase its distance to



(a) (b)

Fig. C.2: a) Sample from MOT17-04 [6]. The yellow boxes illustrate objects partly occluded by scene-occlusion (red) and inter-object-occlusion (blue). b) The blue object displays nearly linear motion, whereas the red object is behaving erratically. The ellipsoids symbolize the confidence of an artificial underlying motion model.

the camera, or the illumination conditions may change. Aside from the visual cues, the object's position is also critical. Intuitively, it becomes less likely to confuse objects as the spatial distance between them increases.

Inter-object-occlusion is typically the most difficult to handle, especially if the objects are of the same type, as the trajectories of multiple objects cross. An example can be seen in Figure C.2a, where the blue box marks a person that partially occludes another person.

4 The MOTCOM Metrics

We propose individual metrics to describe the level of occlusion, erratic motion, and visual similarity for MOT sequences. Subsequently, we combine these three sub-metrics into a higher-level metric that describes the overall complexity of the sequences.

Preliminaries

We define a MOT sequence as a set of frames $F = \{1, 2, \dots\}$ containing a set of objects $K = \{k_1, k_2, \dots\}$. The objects do not have to be present in every frame, therefore, we define the set of frames where a given object is present by $F^k = \{t_1, t_2, \dots\}$. The objects present in a given frame t are defined as the set $K^t = \{k | k \in K \wedge t \in F^k\}$. At each frame t an object k is represented by its center-position in image coordinates and the height and width of the surrounding bounding box $k_t = (x, y, h, w)$.

4.1 Occlusion Metric

As mentioned in Section 3, occlusion can be divided into three types: self-, scene- and inter-object occlusion. In order to quantify the occlusion rate in a sequence, one should ideally account for all three types. However, it is most often non-trivial to determine the level of self-occlusion and it is commonly not taken into account in MOT. Pedersen et al. [13] used the ratio of intersecting object bounding boxes to determine the inter-object occlusion rate. Similarly, the MOT16, MOT17, and MOT20 datasets include a visibility score based on the intersection over area (IoA) of both inter- and scene-objects [7], where IoA is formulated as the area of intersection over the area of the target.

Following this trend, we omit self-occlusion and base the occlusion metric, OCOM, on the IoA and compute it as

$$\text{OCOM} = \frac{1}{|K|} \sum_{k}^{K} \bar{v}^k, \tag{C.1}$$

where $\bar{v}^k$ is the mean level of occlusion of object k. v_t^k is in the interval $[0, 1]$ where 0 is fully visible and 1 is fully occluded. It is assumed that terrestrial objects move on a ground plane which allows us to interpret their y-values as pseudo-depth and decide on the ordering. Annotations are needed to calculate the occlusion level for objects moving in 3D. OCOM is defined in the interval $[0, 1]$ where a higher value means more occlusion and a harder problem to solve.

4.2 Motion Metric

The proposed motion metric, MCOM, is based on the assumption that objects move linearly when observed at small time steps. If this assumption is not upheld, it is a sign of erratic motion and thereby a more complex MOT sequence.

Initially, the displacement vector, P_t^k, between the object's position in the current and past time step is calculated as

$$P_t^k = p_t^k - p_{t-\beta}^k, \tag{C.2}$$

where p_t is the position of object k at time t, defined by its x- and y-coordinates, and β describes the temporal step size. When calculating the displacement between two consecutive frames $\beta = 1$. The displacement vector in the first frame of a trajectory is set to zero and β is capped by the first and last frame of a trajectory when the object is not present at time $t \pm \beta$.

The position in the next time step is predicted using a linear motion model with constant velocity based on the current position and the calculated displacement vector. The position is predicted by

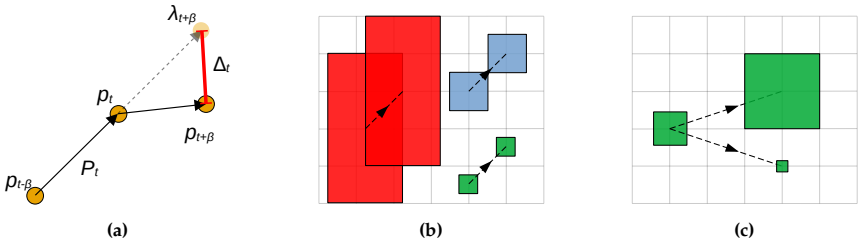


Fig. C.3: a) Illustrative example of how the positional error Δ_t is calculated as the distance between the true position $p_{t+\beta}$ and estimated position $\lambda_{t+\beta}$. b) The three objects have traveled an equal distance. Relative to their size, the two smaller objects are displaced by a larger amount and the bounding box overlap disappears. c) If the size of an object increases between two time steps the displacement is relatively less important, compared to when the size of the object decreases.

$$\lambda_{t+\beta}^k = P_t^k + p_t^k. \tag{C.3}$$

The error between the predicted and true position of the object is calculated by

$$\Delta_t^k = \ell_2(p_{t+\beta}^k, \lambda_{t+\beta}^k) \tag{C.4}$$

where ℓ_2 is the Euclidean distance function and a larger Δ_t^k indicates a more complex motion. See Figure C.3a for an illustration of how the displacement error is calculated. This approach may seem overly simplified, but it encapsulates changes in both direction and velocity. Furthermore, it is deliberately sensitive to low frame rates and camera motion, as both factors add to the complexity of tracking sequences.

Inspired by the analysis of decreasing tracking performance with respect to smaller object sizes by Bergmann et al. [12], the size is also taken into consideration. The combination of size and movement affects the difficulty of predicting the next state of the object. In Figure C.3b, the rectangles are equally displaced but do not experience the same displacement relative to their size. Intuitively, if a set of objects are moving at similar speeds, it is harder to track the smaller objects due to their lower spatio-temporal overlap.

Accordingly, the motion-based complexity measure is based on the

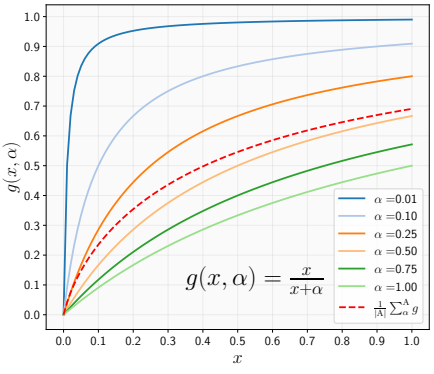


Fig. C.4: α controls the growth of the function $g(x, \alpha)$ and decides when an output value of 0.5 is reached. The dashed line illustrates $g(x, \alpha)$ when using the average of a set of α values.

displacement relative to the size of the object. As illustrated in Figure C.3c, the size of the object may change between two time steps. The direction of the change is critical as the displacement is less distinct if the size of the object is increasing, compared to the opposite situation. Therefore, we multiply the current size of the object with the change in object size to get the transformed object size

$$\rho_t^k = s_t^k \cdot \frac{s_{t+\beta}^k}{s_t^k} = s_{t+\beta}^k, \tag{C.5}$$

where $s_t^k = \sqrt{w_t^k \cdot h_t^k}$ and h_t^k and w_t^k are the height and width of object k at time step t, respectively. The motion complexity measure is then calculated as the mean size-compensated displacement across all frames, F, and all objects at each frame, K^t, and weighted by the log-sigmoid function $g(x, \alpha)$

$$\text{MCOM} = \frac{1}{|A|} \sum_{\alpha}^{A} g\left(\frac{1}{\sum_k^K |F^k|} \sum_k^K \sum_t^{F^k} \frac{\Delta_t^k}{\rho_t^k}, \alpha \right), \tag{C.6}$$

where the average of $A = \{0.01, 0.02, ..., 1.0\}$ is used to avoid manually deciding on a specific value for α. The use of the function $g(x, \alpha)$ is motivated by the aim of having an output in the range $[0, 1]$, where a higher number describes a more complex motion. The function $g(x, \alpha)$ is given by

$$g(x, \alpha) = \frac{1}{1 + e^{-\log(x)}\alpha} = \frac{1}{1 + \frac{\alpha}{x}} = \frac{x}{x + \alpha}, \tag{C.7}$$

where α affects the gradient of the monotonically increasing function and indicates the point where the output of the function will reach 0.5 as illustrated in Figure C.4. The function is designed such that displacements in the lower ranges are weighted higher. The argument for this choice is based on the assumption that minor increments to an extraordinarily erratic locomotive behavior have less impact on the complexity.

4.3 Visual Similarity Metric

In order to define a metric that links an object's visual appearance with tracking complexity, we investigate how similar an object in one frame is compared to itself and other objects in the next frame. Two objects may look similar, but they cannot occupy the same spatial position. Therefore, we propose a spatial-aware visual similarity metric called VCOM.

VCOM consists of a preprocessing, feature extraction, and distance evaluation step. For every object $k \in K$ in every frame $t \in F$ an image I_t^k is produced with the object's bounding box in focus and a heavy blurred background. We

blur the image using a discrete Gaussian function, except in the region of the object's bounding box as visualized in Figure C.5a.

A feature embedding is then extracted from each of the preprocessed images. As opposed to looking at the bounding box alone, using the entire image allows us to retain and embed spatial information in the feature vector. The object's location is especially valuable in scenes with similarly looking objects and the blurred background contributes with low frequency information of the surroundings.

We blur the image with a Gaussian kernel with a fixed size of 201 and a sigma of 38 and extract the image features using an ImageNet [44] pre-trained ResNet-18 [45] model. We measure the similarity between the feature vector of the target object in frame t and the feature vectors of all the objects in frame $t+1$ by computing the Euclidean distance. The uncertainty increases if more objects are located within the proximity of the target. Therefore, we do not only look for the nearest neighbor, but rather the number of objects within a given distance, $d(r)$, from the target feature vector

$$d(r) = d_{\text{NN}} + d_{\text{NN}} \cdot r \tag{C.8}$$

where d_{NN} is the distance to the nearest neighbor and r is a distance ratio. The ratio is multiplied by the distance to the nearest neighbor in order to account for the variance in scale, e.g., as induced by object resolution or distinctiveness.

An object within the distance boundary that shares the same identity as the target object is considered a true positive (TP) and all other objects are considered false positives (FP). By measuring the complexity based on the false discovery rate, FDR $= \frac{\text{FP}}{\text{FP+TP}}$, we get an output in the range $[0, 1]$ where

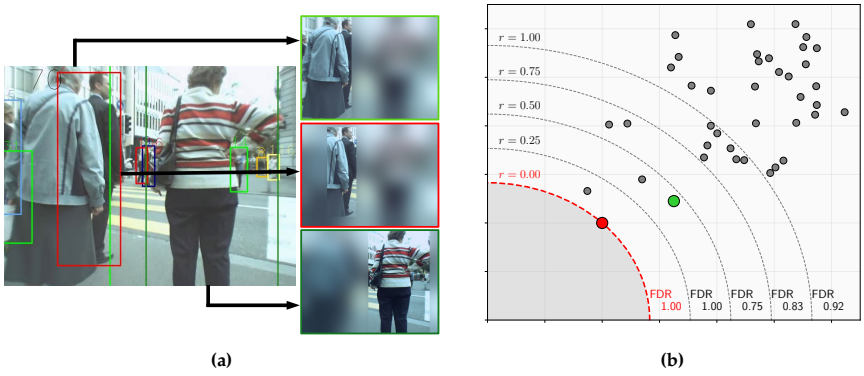


Fig. C.5: a) Example showing three images with the object in focus and a blurred background produced from a frame from the MOT17-05 sequence. b) The distance ratio, r, affects the FDR when other objects are in the proximity of the target. The red dot is the nearest neighbor, the green dot is the true positive match, and the remaining dots are other objects.

a higher number indicates a more complex task. An illustrative example of how the FDR is determined based on the distance ratio r can be seen in Figure C.5b. It is ambiguous to choose a single optimal distance ratio r. Therefore, we calculate VCOM based on the average of distance ratios from the set $R = \{0.01, 0.02, ..., 1.0\}$

$$\text{VCOM} = \frac{1}{|R|} \sum_{r}^{R} \frac{1}{|F|} \sum_{t}^{F} \frac{1}{|K^t|} \sum_{k}^{K^t} FDR_{d(r)}(k) \tag{C.9}$$

4.4 MOTCOM

Occlusion alone does not necessarily indicate an overwhelming problem if the object follows a known motion model or if it is visually distinct. The same is true for erratic motion and visual similarity when viewed in isolation. However, the combination of occlusion, erratic motion, and visual similarity becomes increasingly difficult to handle.

Therefore, we combine the occlusion, erratic motion, and visual similarity metrics into a single MOTCOM metric that describes the overall complexity of a sequence. MOTCOM is computed as the weighted arithmetic mean of the three sub-metrics and is given by

$$\text{MOTCOM} = \frac{w_{\text{OCOM}} \cdot \text{OCOM} + w_{\text{MCOM}} \cdot \text{MCOM} + w_{\text{VCOM}} \cdot \text{VCOM}}{w_{\text{OCOM}} + w_{\text{MCOM}} + w_{\text{VCOM}}} \tag{C.10}$$

where w_{OCOM}, w_{MCOM}, and w_{VCOM} are the weights for the three sub-metrics. Equal weighting can be obtained by setting $w_{\text{OCOM}} = w_{\text{MCOM}} = w_{\text{VCOM}}$, while custom weights may be suitable for specific applications. During evaluation we weight the sub-metrics equally as we deem each of the sub-problems equally difficult to handle.

5 Evaluation

In the following experimental section, we demonstrate that MOTCOM is able to describe the complexity of MOT sequences and is superior to *density* and *number of tracks*. In order to do this, we compare the estimated complexity levels with ground truth representations. Such ground truths are not readily available, but a strong proxy can be obtained by ranking the sequences based on the performance of state-of-the-art trackers [46]. There exist many performance metrics with two of the most popular being MOTA [47] and IDF1 [48]. However, we apply the recent HOTA metric [10], which was proposed in response to the imbalance between detection, association, and localization within traditional metrics. Additionally, HOTA is the tracker performance metric that correlates the strongest with MOT complexity based on human

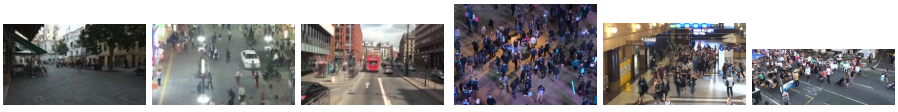
assessment [10]. In the remainder of this section, we present the datasets and evaluation metrics we use to experimentally verify the applicability of MOTCOM.

5.1 Ground Truth

In order to create a strong foundation for the evaluation, we are in need of benchmark datasets with consistent annotation standards and leader boards with a wide range of state-of-the-art trackers. Therefore, we evaluate MOTCOM on the popular MOT17 [6] and MOT20 [7] datasets[1]. There are seven sequences in the test split of MOT17 and four sequences in the test split of MOT20, some of which are presented in Figure C.6. Furthermore, leader boards are provided for both benchmarks with results from 212 trackers for MOT17 and 80 trackers for MOT20. We use the results from the top-30 ranked trackers[2] based on the average HOTA score, so as to limit unstable and fluctuating performances.

In order to strengthen and support the evaluation, we include the training split of the fully synthetic MOTSynth dataset [11] which contains 764 varied sequences of pedestrians. A few samples from the dataset can be seen in Figure C.7. In order to obtain ground truth tracker performance for MOTSynth, we train and test a CenterTrack model [20] on the data. We have chosen CenterTrack as it has been shown to perform well when trained on synthetic data [11].

[1]With permission from the MOTChallenge benchmark authors.
[2]Leader board results obtained on March 4, 2022.



(a) (b)

Fig. C.6: Sample images from a) MOT17 [6] and b) MOT20 [7]. MOT17 contains varied urban scenes with and without camera motion. MOT20 contains crowded scenes captured from an elevated point of view and without camera motion.

Fig. C.7: Sample images from the MOTSynth dataset [11]. The sequences vary in camera motion and perspective, environment, and lighting.

5.2 Evaluation Metrics

We evaluate and compare the dataset complexity metrics by their ability to rank the MOT sequences according to the HOTA score of the trackers. We rank the sequences from simple to complex by their *density*, *number of tracks* (abbr. *tracks*), MOTCOM score, and HOTA score. Depending on the metric, the ranking is in decreasing (HOTA) or increasing order (*density*, *tracks*, MOTCOM). The absolute difference between the ranks, known as Spearman's Footrule Distance (FD) [49], gives the distance between the ground truth and estimated ranks

$$\mathrm{FD} = \sum_{i=1}^{n} |\mathrm{rank}(x_i) - \mathrm{rank}(\mathrm{HOTA}_i)|, \tag{C.11}$$

where n is the number of sequences and x is *density*, *tracks*, or MOTCOM. In order to directly compare results of sets of different lengths, we normalize the FD by the maximal possible distance $\mathrm{FD}_{\max}$ which is computed as

$$\mathrm{FD}_{\max} = \begin{cases} \sum_{i=1}^{n} i - \frac{n}{2} & \{n \mid 2m\ ,\ m \in \mathbb{Z}^+\} \\ \sum_{i=1}^{n} i - \frac{n+1}{2} & \{n \mid 2m-1\ ,\ m \in \mathbb{Z}^+\} \end{cases}. \tag{C.12}$$

Finally, we compute the normalized FD, $\mathrm{NFD} = \frac{\mathrm{FD}}{\mathrm{FD}_{\max}}$.

6 Results

In Table C.1, we present the mean FD of the ranks of *density*, *tracks*, and MOTCOM against the ground truth ranks dictated by the average top-30 HOTA performance on the MOT17 and MOT20 test splits (individually and in combination). The numbers in parentheses are the normalized FD. Generally, MOTCOM has a considerably lower FD compared to *density* and *tracks*. TIn Table C.1, we present the mean FD of the ranks of *density*, *tracks*, and MOTCOM against the ground truth ranks dictated by the average top-30 HOTA performance on the MOT17 and MOT20 test splits (individually and in combination). The numbers in parentheses are the normalized FD. Generally,

Table C.1: Ground truth ranks are based on the average top-30 HOTA performance. The results are presented as the mean FD and the NFD in parentheses. A lower score is better and the results in bold are the lowest

Top-30	$\mathrm{MOT17}_{\mathrm{test}}$	$\mathrm{MOT20}_{\mathrm{test}}$	Combined
Density	1.71 (0.50)	1.00 (0.50)	3.82 (0.70)
Tracks	2.57 (0.75)	1.50 (0.75)	3.82 (0.70)
MOTCOM	**0.86 (0.25)**	**0.00 (0.00)**	**1.45 (0.27)**

Table C.2: Ground truth ranks are based on the CenterTrack HOTA performance. The results are presented as the mean FD and the NFD in parentheses. A lower score is better and the results in bold are the lowest

CenterTrack	$\text{MOTChl}_{\text{test}}$	$\text{MOTChl}_{\text{train}}$	$\text{MOTChl}_{\text{both}}$	MOTSynth
Density	3.27 (0.60)	4.18 (0.77)	7.36 (0.67)	238.71 (0.63)
Tracks	2.73 (0.50)	3.64 (0.67)	6.64 (0.60)	193.50 (0.51)
MOTCOM	**2.36 (0.43)**	**2.18 (0.40)**	**4.82 (0.44)**	**100.17 (0.26)**

MOTCOM has a considerably lower FD compared to *density* and *tracks*. This suggests that MOTCOM is better at ranking the sequences according to the HOTA performance.

A similar tendency can be seen for the CenterTrack-based results presented in Table C.2. In order to increase the number of samples, we have evaluated CenterTrack on both the train and test splits of the MOT17 and MOT20 datasets. $\text{MOTChl}_{\text{test}}$ and $\text{MOTChl}_{\text{train}}$ are the test and train sequences, respectively, of MOT17 *and* MOT20. $\text{MOTChl}_{\text{both}}$ includes *all* the sequences from MOT17 and MOT20. These results support our claim that MOTCOM is better at estimating the complexity of MOT sequences compared to *density* and *tracks*.

We present a Spearman's correlation matrix in Figure C.8 based on the top-30 trackers evaluated for the combined MOT17 and MOT20 test splits. It indicates that the *density* and *tracks* do not correlate with HOTA, MOTA, or IDF1, whereas MOTCOM has a strong negative correlation with all the performance metrics. Trackers evaluated on sequences with high MOTCOM scores tend to have lower performance while sequences with low MOTCOM scores gives higher performance. This underlines that MOTCOM can indeed be used to understand the complexity of MOT sequences.

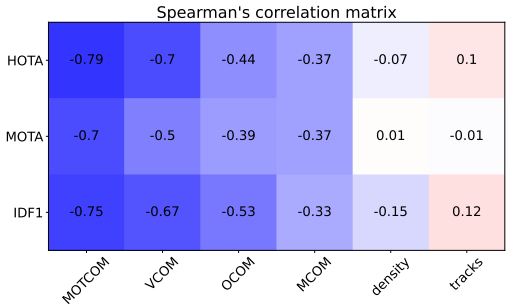


Fig. C.8: Spearman's correlation matrix based on the performance of the top-30 trackers on MOT17 and MOT20.

7 Discussion

Our complexity metric MOTCOM provides tracker researchers and dataset developers a comprehensive score to investigate and describe the complexity of MOT sequences without the need for multiple baseline evaluations of dif-

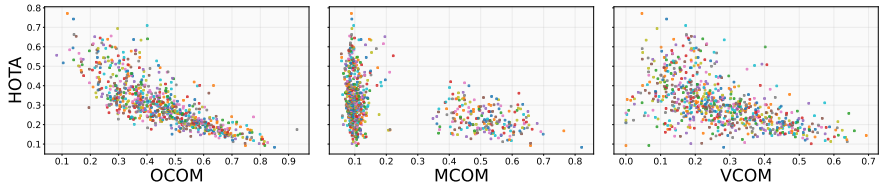


Fig. C.9: The CenterTrack-based HOTA scores of the MOTSynth sequences plotted against the sub-metrics OCOM, MCOM, and VCOM, respectively.

ferent tracking methods. This allows for an objective comparison of different datasets without introducing potential training bias. Currently, the assessment of tracker performance is roughly speaking reduced to a placement on a benchmark leader board. This underrates novel solutions developed for less popular datasets or methods designed explicitly to solve sub-tasks such as occlusion or erratic motion.

Supplemented by the sub-metrics, MOTCOM provides a deeper understanding and more informed discussions on dataset composition and tracker performance, which will increase the explanability of MOT. In order to illustrate this, we discuss the performance of CenterTrack on the MOTSynth dataset with respect to MOTCOM. Here we see that the occlusion level (OCOM) in Figure C.9 has a strong negative correlation with the HOTA score and the visual similarity metric (VCOM) has a relatively weak correlation with HOTA. Both cases expose the design of CenterTrack, which does not contain a module to handle lost tracks and is not dependent on visual cues for tracking. For the motion metric (MCOM) we see two distributions; one in the lower end and one in the upper end of the MCOM range. The objects are expected to behave similarly, so this indicates that parts of the MOTSynth sequences include heavy camera motion which is difficult for CenterTrack to handle. In Figure C.10, we show that MOTCOM is far better at estimating the complexity level compared to *tracks* and *density*.

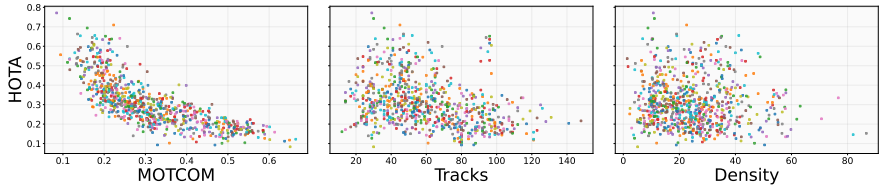


Fig. C.10: The CenterTrack-based HOTA scores of the MOTSynth sequences plotted against MOTCOM, *tracks*, and *density*.

8 Conclusion

We propose MOTCOM, the first meaningful and descriptive MOT dataset complexity metric, and show that it is preferable for describing the complexity of MOT sequences compared to the conventional methods of *number of tracks* and *density*. MOTCOM is a combination of three individual sub-metrics that describe the complexity of MOT sequences with respect to key obstacles in MOT: occlusion, erratic motion, and visual similarity. The information provided by MOTCOM can assist tracking researchers and dataset developers in acquiring a deeper understanding of MOT sequences and trackers. We strongly suggest that the community uses MOTCOM as the prevalent complexity measure for increasing the explainability of MOT trackers and datasets.

Acknowledgements

This work has been funded by the Independent Research Fund Denmark under case number 9131-00128B.

C Supplementary Material

Computing MOTCOM.

We have evaluated four averaging methods for combining the sub-metrics into MOTCOM. The four methods are the arithmetic, quadratic, geometric, and harmonic means and they are presented in Equation (13), Equation (14),Equation (15), and Equation (16), respectively.

$$\text{arithmetic} = \frac{1}{n}\sum_{i=1}^{n} m_i \qquad (13)$$

$$\text{quadratic} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} m_i^2} \qquad (14)$$

$$\text{geometric} = \sqrt[n]{\prod_{i=1}^{n} m_i} \qquad (15)$$

$$\text{harmonic} = \frac{n}{\sum_{i=1}^{n} \frac{1}{m_i}} \qquad (16)$$

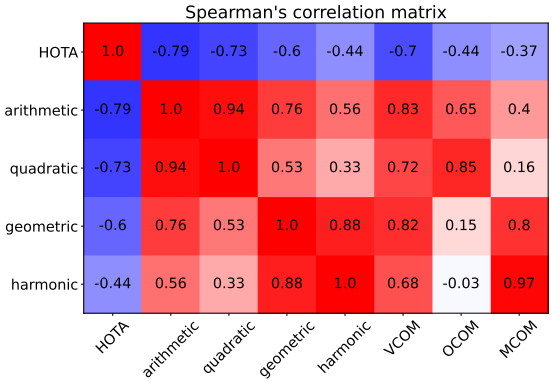
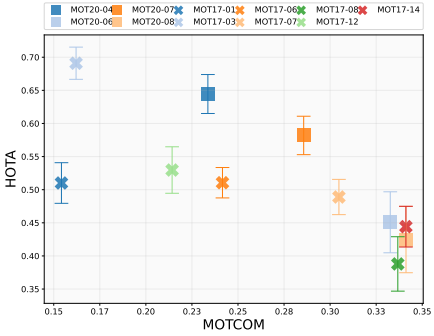


Fig. 11: Spearman's correlation matrix. The entries represent the MOTCOM values when the sub-metrics are combined using the four different averaging methods. The HOTA performance is the average of the top-30 ranked trackers. The scores are based on the combined MOT17 and MOT20 test splits.

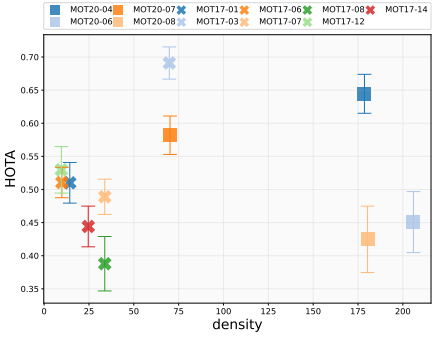
We present the four variations of MOTCOM in Figure 11 computed on the combined MOT17 and MOT20 test splits. We see that they all correlate negatively with the HOTA score. However, the arithmetic mean has the strongest negative correlation and it correlates positively with all the sub-metrics. Therefore, we suggest to compute MOTCOM as the arithmetic mean of the sub-metrics.

Complexity Score Plots for MOT17 and MOT20

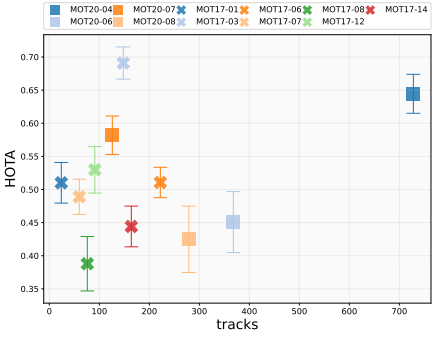
In the main paper we evaluate MOTCOM, *density*, and *tracks* on the MOT17 and MOT20 test splits. We focus mainly on the ranking capabilities of the metrics as we expect tracker performance to have a monotonic, but not necessarily linear, relationship with complexity. The ranks of MOTCOM, *density*, and *tracks* presented in the main paper are based on the scores displayed in Figure 12.



(a) MOTCOM vs. HOTA.



(b) Density vs. HOTA.



(c) Tracks vs. HOTA.

Fig. 12: Average HOTA performance of the top-30 trackers on MOT17 and MOT20 test split against a) MOTCOM, b) *density*, and c) *tracks*. Square markers represent MOT20 sequences and crosses are MOT17 sequences.

The position of the marker indicates the average score and the error bar is the standard deviation. The marker of the MOT17 sequences is a cross and the MOT20 sequences are represented by a square. We see that the MOT20 sequences have significantly higher densities and more tracks compared to the MOT17 sequences, but the HOTA performance is not correspondingly low. This illustrates that *density* and *tracks* do not suffice to describe the complexity of MOT sequences.

Complete Spearman's Correlation Matrix for MOT17 and MOT20.

In Figure 9 in the main paper we presented a partial Spearman's correlation matrix based on the MOT17 and MOT20 sequences. We used the matrix to evaluate the monotonic relationship between the three complexity metrics (MOTCOM, *density*, and *tracks*) and HOTA, MOTA, and IDF1. In Figure 13 we present the complete Spearman's correlation matrix, which shows additional details on the relationship between the entries.

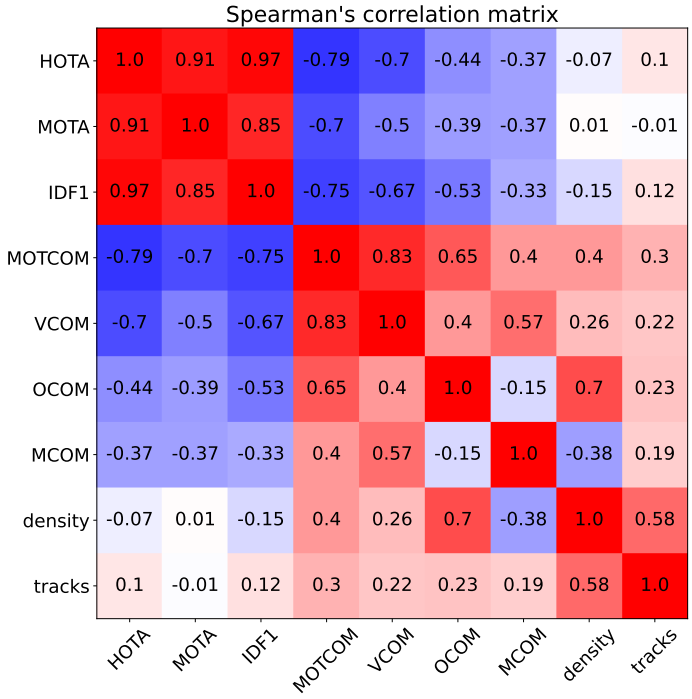


Fig. 13: Spearman's correlation matrix. Based on the average performance of the top-30 trackers on MOT17 and MOT20 test split.

Complete Spearman's Correlation Matrix for MOTSynth.

In the main paper we presented the Spearman's Footrule Distance and the complexity scores for the MOTSynth sequences. To expand upon this, we include the complete Spearman's correlation matrix for the MOTSynth train split in Figure 14. The matrix gives a detailed overview of the monotonic relationship between the entries.

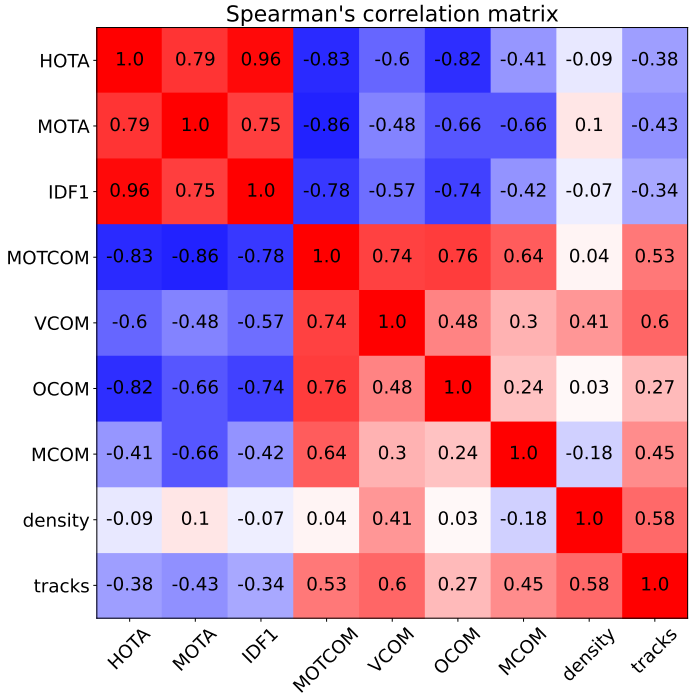


Fig. 14: Spearman's correlation matrix. Based on the CenterTrack performance on the MOTSynth train split.

References

[1] J. K. Uhlmann, "Algorithms for multiple-target tracking," *American Scientist*, vol. 80, no. 2, pp. 128–141, 1992.

[2] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial Intelligence*, vol. 293, p. 103448, 2021.

[3] R. Gade and T. B. Moeslund, "Constrained multi-target tracking for team sports activities," *IPSJ Transactions on Computer Vision and Applications*, vol. 10, p. 2, 2018.

[4] A. Pérez-Escudero, J. Vicente-Page, R. C. Hinz, S. Arganda, and G. G. De Polavieja, "idtracker: tracking individuals in a group by automatic identification of unmarked animals," *Nature methods*, vol. 11, no. 7, pp. 743–748, 2014.

[5] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2012, pp. 3354–3361.

[6] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv:1603.00831*, mar 2016.

[7] P. Dendorfer, A. Osep, A. Milan, K. Schindler, D. Cremers, I. Reid, S. Roth, and L. Leal-Taixé, "Motchallenge: A benchmark for single-camera multiple target tracking," *International Journal of Computer Vision (IJCV)*, vol. 129, p. 845–881, 2021.

[8] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2443–2451.

[9] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[10] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *International Journal of Computer Vision (IJCV)*, vol. 129, no. 2, p. 548–578, oct 2021.

[11] M. Fabbri, G. Brasó, G. Maugeri, O. Cetintas, R. Gasparini, A. Ošep, S. Calderara, L. Leal-Taixe, and R. Cucchiara, "Motsynth: How can synthetic data help pedestrian detection and tracking?" in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 10 829–10 839.

[12] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2019, pp. 941–951.

[13] M. Pedersen, J. B. Haurum, S. Hein Bengtson, and T. B. Moeslund, "3d-zef: A 3d zebrafish tracking benchmark dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2423–2433.

[14] A. Andriyenko, S. Roth, and K. Schindler, "An analytical formulation of global occlusion reasoning for multi-target tracking," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*. IEEE, 2011, pp. 1839–1846.

[15] A. Andriyenko and K. Schindler, "Multi-target tracking by continuous energy minimization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1265–1272.

[16] J. Xu, Y. Cao, Z. Zhang, and H. Hu, "Spatial-temporal relation networks for multi-object tracking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 3987–3997.

[17] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3464–3468.

[18] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 3645–3649.

[19] J. Peng, C. Wang, F. Wan, Y. Wu, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu, "Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking," in *Computer Vision – ECCV 2020*. Cham: Springer, 2020, pp. 145–161.

[20] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Computer Vision – ECCV 2020*. Springer, 2020, pp. 474–490.

[21] J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell, and F. Yu, "Quasi-dense similarity learning for multiple object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 164–173.

[22] Z. Lu, V. Rathod, R. Votel, and J. Huang, "Retinatrack: Online single stage joint detection and tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 14 656–14 666.

[23] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *International Journal of Computer Vision (IJCV)*, vol. 129, no. 11, pp. 3069–3087, Sep. 2021.

[24] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 36, no. 1, pp. 58–72, 2014.

[25] D. Stadler and J. Beyerer, "Improving multiple pedestrian tracking by track management and occlusion handling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 10 953–10 962.

[26] X. Cao, S. Guo, J. Lin, W. Zhang, and M. Liao, "Online tracking of ants based on deep association metrics: method, dataset and evaluation," *Pattern Recognition*, vol. 103, 2020.

[27] C. Liu, R. Yao, S. H. Rezatofighi, I. Reid, and Q. Shi, "Model-free tracker for multiple objects using joint appearance and motion inference," *IEEE Transactions on Image Processing (TIP)*, vol. 29, pp. 277–288, 2020.

[28] W. Luo, T.-K. Kim, B. Stenger, X. Zhao, and R. Cipolla, "Bi-label propagation for generic multiple object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1290–1297.

[29] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 261–268.

[30] L. Kratz and K. Nishino, "Tracking with local spatio-temporal motion patterns in extremely crowded scenes," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 693–700.

[31] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. C. H. Hoi, "Deep learning for person re-identification: A survey and outlook," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 44, no. 6, pp. 2872–2893, jun 2022.

[32] S. D. Khan and H. Ullah, "A survey of advances in vision-based vehicle re-identification," *Computer Vision and Image Understanding*, vol. 182, pp. 50–63, 2019.

[33] S. Schneider, G. W. Taylor, S. Linquist, and S. C. Kremer, "Past, present and future approaches using computer vision for animal re-identification from camera trap data," *Methods in Ecology and Evolution*, vol. 10, no. 4, pp. 461–470, jan 2019.

[34] J. Bruslund Haurum, A. Karpova, M. Pedersen, S. Hein Bengtson, and T. B. Moeslund, "Re-identification of zebrafish using metric learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, 2020, pp. 1–11.

[35] S. Schneider, G. W. Taylor, and S. C. Kremer, "Similarity learning networks for animal individual re-identification - beyond the capabilities of a human observer," in *Proceedings of the IEEE Winter Applications of Computer Vision Workshops (WACVW)*, 2020, pp. 44–52.

[36] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler, "Learning by tracking: Siamese cnn for robust target association," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016, pp. 418–425.

[37] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4705–4713.

[38] J. Yin, W. Wang, Q. Meng, R. Yang, and J. Shen, "A unified object motion and affinity model for online multi-object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 6767–6776.

[39] T. K. Ho and M. Basu, "Complexity measures of supervised classification problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 24, no. 3, pp. 289–300, 2002.

[40] F. Branchaud-Charron, A. Achkar, and P.-M. Jodoin, "Spectral metric for dataset complexity assessment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3210–3219.

[41] Y. Cui, Z. Gu, D. Mahajan, L. van der Maaten, S. Belongie, and S.-N. Lim, "Measuring dataset granularity," 2019.

[42] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "MOTS: multi-object tracking and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7934–7943.

[43] A. Milan, K. Schindler, and S. Roth, "Challenges of ground truth evaluation of multi-target tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2013, pp. 735–742.

[44] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2009, pp. 248–255.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016, pp. 770–778.

[46] L. Leal-Taixé, A. Milan, K. Schindler, D. Cremers, I. Reid, and S. Roth, "Tracking the trackers: an analysis of the state of the art in multiple object tracking," *arXiv:1704.02781*, 2017.

[47] R. Stiefelhagen, K. Bernardin, R. Bowers, J. Garofolo, D. Mostefa, and P. Soundararajan, "The clear 2006 evaluation," in *Multimodal Technologies for Perception of Humans*. Berlin, Heidelberg: Springer, 2007, pp. 1–44.

[48] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Computer Vision – ECCV 2016 Workshops*. Springer, 2016, pp. 17–35.

[49] P. Diaconis and R. L. Graham, "Spearman's footrule as a measure of disarray," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 2, pp. 262–268, 1977.

Paper D

Detection of Marine Animals in a New Underwater Dataset with Varying Visibility

Malte Pedersen, Joakim Bruslund Haurum,
Rikke Gade, and Thomas B. Moeslund

The paper has been published in the
Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 18-26, 2019.

Malte Pedersen, Joakim Bruslund Haurum, Rikke Gade, Niels Madsen, and Thomas B. Moeslund, ”Detection of Marine Animals in a New Underwater Dataset with Varying Visibility". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.

The layout has been revised.

Abstract

The increasing demand for marine monitoring calls for robust automated systems to support researchers in gathering information from marine ecosystems. This includes computer vision based marine organism detection and species classification systems. Current state-of-the-art marine vision systems are based on CNNs, which in nature require a relatively large amount of varied training data. In this paper we present a new publicly available underwater dataset with annotated image sequences of fish, crabs, and starfish captured in brackish water with varying visibility. The dataset is called the Brackish Dataset and it is the first part of a planned long term monitoring of the marine species visiting the strait where the cameras are permanently mounted. To the best of our knowledge, this is the first annotated underwater image dataset captured in temperate brackish waters. In order to obtain a baseline performance for future reference, the YOLOv2 and YOLOv3 CNNs were fine-tuned and tested on the Brackish Dataset.

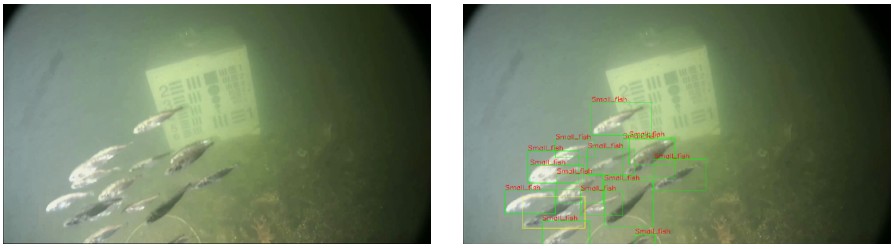
1 Introduction

More than 70% of the Earth is covered by water and our oceans plays a vital role for humans all around the globe. In order to reduce declination of biodiversity and uphold sustainable fisheries, it is important to keep our oceans healthy. A necessary step towards a better understanding of marine life and ecosystems is to monitor and analyze the impact human activities have on our waters both on a local, regional, and international scale [1].

As underwater cameras and technology in general become more accessible, automatic computer vision based methods are being developed for efficient detection and classification of marine animals and plants, which can be of great aid for marine researchers in analyzing and monitoring our oceans. While underwater images captured in pure water can be handled much like regular images captured above water, other factors must be taken into account when processing images from natural waters. The optical properties of natural water depend on the absorption and scattering of light. While the light scattering in pure water only depends on the temperature and pressure, natural waters show much larger temporal and spatial variations due to the varying content of dissolved and particulate matter. These particles affect both scattering and absorption of light and are often visible as noise in the images [2, 3].

Within scientific communities, it is common practice to evaluate the performance of methods on the same dataset to allow for fair comparison and benchmarking. However, to be able to develop robust algorithms for, e.g., analysis of marine life and ecosystems, the datasets need to represent the natural variations in optical properties seen in natural waters across the world.

To the best of our knowledge, there exists no large scale labeled dataset of



(a) Frame from a sequence with a school of *small fish*.

(b) Same frame as above, but with annotations drawn on the image.

Fig. D.1: A frame example from the proposed dataset. The same frame is shown with and without annotations.

temperate coastal or estuarine environments that allows for the development of methods for detecting and classifying marine species in such waters. In particular, none of the publicly available datasets are captured in European marine environments. This is needed to develop robust methods for all water types and marine species, and furthermore, it is needed in order to reach the goal of Good Environmental Status (GES), stipulated by the European Union's Marine Strategy Framework Directive (MSFD) [4, 5].

With this paper, we strive to accommodate this need by releasing a publicly available dataset that has been captured over several weeks in temperate brackish water. Hence, it includes natural variation derived from, e.g., time of day, weather conditions, and activities in the water. The dataset is the first stage of a long-term marine monitoring project where three cameras continuously capture video of the marine life near the bottom of Limfjorden in Denmark, nine meters below the water surface. The aim of this paper is to present this new annotated dataset which, due to its uniqueness, is an important addition to existing annotated marine image datasets. Furthermore, we evaluate two existing state-of-the-art detection methods on the new dataset and present the results as a baseline for future reference.

1.1 Contributions

- A publicly available underwater dataset[1] containing bounding box annotated sequences of images containing *big fish*, *small fish*, *starfish*, *shrimps*, *jellyfish*, and *crabs* captured in a brackish strait with varying visibility.
- An overview of annotated underwater image datasets.
- A baseline evaluation of state-of-the-art detection methods on the presented dataset.

[1] `https://www.kaggle.com/aalborguniversity/brackish-dataset`

An example from the proposed dataset can be seen in Figure D.1, where a frame from a sequence with a school of *small fish* is presented with and without annotations.

2 Related Work

As the oceans cover everything from the dark abyssal zone to the sunny coral reefs, the variation between underwater datasets and their associated detection methods can be significant. This section presents an overview of some of the areas where computer vision algorithms have been developed to assist in the huge task of monitoring our oceans and an overview of annotated marine image datasets.

The term *marine vision* will be used in the remainder of this paper as an umbrella term covering the methods and algorithms developed for the purpose of assisting in monitoring marine environments.

2.1 Marine Vision Methods

Detection and classification of fish has been addressed by Villon et al. [6] who investigates the performance of a traditional Support Vector Machines (SVM) classifier trained on Histogram of Oriented Gradients (HOG) features for classifying coral reef fish and compares it with the performance of a fine-tuned Convolutional Neural Network (CNN). Their tests show that the CNN outperforms the traditional classification methods. The same conclusion is reached by Salman et al. [7] who compares traditional classification methods such as SVM, k-Nearest Neighbours (k-NN), and Sparse Representation Classifier (SRC) with CNN. They achieve an average classification rate of more than 90% on the LifeCLEF14 [8] and LifeCLEF15 [9] fish datasets using CNN and generally a significantly lower rate using the traditional methods. Siddiqui et al. [10] reaches state-of-the-art performance on fish species classification using a very deep CNN with a cross-layer pooling approach for enhanced discriminative ability in order to handle the problem of limited labelled training data.

Another interesting marine vision area is scallop detection which has been investigated by Dawkins and Gallager [11]. Using multiple features and a series of cascaded Adaboost classifiers, they developed one of the most prominent scallop detection algorithms. A more recent attempt was presented by Rasmussen et al. [12] who tested variations of the YOLOv2 CNN trained for scallop detection. They achieved high accuracy while being able to run in real-time for keeping up with live recordings from an Autonomous Underwater Vehicle (AUV).

Coral reefs are of great interest to marine biologists worldwide but they are difficult and tedious to monitor. In order to assist biologists, Mahmood et

al. [13] fine-tuned a VGGNet using a subset of the Benthoz-15 dataset [14] and used it for automatically analyzing the coral coverage of three sites in Western Australia. Another approach was investigated by Beijbom et al. [15] who achieved state-of-the-art performance by fusing standard reflectance images with fluorescence images of corals in a 5 channel CNN.

2.2 Annotated Marine Image Datasets

A thing that is common for state-of-the-art detection methods, and deep learning methods in particular, is that they are in need of relatively large amounts of training data. One of the most popular underwater datasets for fish detection and species classification is the F4K dataset [16]. It was recorded from 10 cameras between 2010 and 2013 in Taiwan and it has been used for multiple detection and classification algorithms [6, 7, 17–20]. The F4K dataset is large and consists of videos and images with complex scenes, various marine species and lots of annotations making it an obvious benchmark dataset. It was also used as part of the LifeCLEF tasks [8, 9, 21].

Another large dataset is the Jamstec E-Library of Deep-sea Images (J-EDI) [22], which consists of videos and images of deep sea organisms captured by Remotely Operated underwater Vehicles (ROV). The images of the J-EDI dataset are annotated on an image level and have been used to train CNNs for detection of deep sea organisms [23, 24]. Two other datasets with focus on fish are the Croatian Fish Dataset [25] which consists of cropped images of 12 different fish species and the QUT Fish Dataset [26] which consists of fish images both in and out of water.

However, as already mentioned, it is not only fish that are of interest within marine vision. Another critical field is monitoring of benthic organisms, such as scallops and corals. The HabCam dataset [27, 28] consists of 2.5 millions annotated images of mainly scallops, but also fish and starfish. The images have been captured along the continental shelf off the east coast of the USA and was used in the work presented by Dawkins and Gallager [11].

The BENTHOZ-2015 dataset [14] is a benthic dataset recorded along the coasts of Australia and used for classifying corals [13]. The Tasmania Coral Point Count [29] was recorded in 2008 during 22 dive missions using an AUV off the South-East coast of Tasmania and has been used for kelp detection [30].

Other annotated coral reef datasets include the Moorea Labeled Corals [31] which is an annotated subset of the Moorea Labeled Corals Long Term Ecological Research project in French Polynesia and the Eilat Fluorescence dataset [15], which experiments with a combination of standard reflectance and fluorescence images in order to improve the coral classification rate of CNNs. Both datasets have been recorded using a custom variation of a photoquadrat [32].

A collection of datasets has been published for the data challenge of the

workshop "Automated Analysis of Marine Video for Environmental Monitoring" in 2018 and 2019 [33]. These datasets include a part of the HabCam dataset [28], as well as four other datasets, MOUSS, AFSC, MBARI, and NWFSC and the images are annotated with either keypoints or bounding boxes.

A table summarizing the underwater datasets can be found in Table D.1 along with the proposed dataset, named the Brackish Dataset, which will be described in further details in Section 4.

	Environment	Recording type	Visibility	Sensor	Images	Labeling
F4K - Complex [34]	Reef	Stationary	Varying	RGB	14 videos	Bounding box
F4K - Species [35]	Reef	Stationary	Varying	RGB	27,370	Masks
F4K - Trajectories [36]	Reef	Stationary	Varying	RGB	93 videos	Bounding box
J-EDI [22]	Deep sea	ROV	Clear	RGB	1,500,000	Image level
Croatian Fish Dataset [25]	-	Various	Varying	RGB	794	Bounding box
QUT Fish Dataset [26]	-	Various	Clear	RGB	3,960	Bounding box
HabCam [28]	Shelf sea	Towing	Clear	Stereo	2,500,000	Bounding box
Benthoz-15 [14]	Reef	AUV	Clear	Stereo	9,874	Points
Tasmania Coral Point Count [29]	Reef	AUV	Clear	Stereo	1,258	Points
The Moorea Labeled Corals [31]	Reef	Photoquadrat	Clear	RGB	2,055	Points
Eilat Fluorescence [15]	Reef	Photoquadrat	Clear	RGB	212	Points
MOUSS [33]	Ocean floor	Stationary	Clear	Gray	159	Bounding box
AFSC [33]	Ocean	ROV	Clear	RGB	571	Points
MBARI [33]	Ocean floor	-	Clear	RGB	666	Bounding box
NWFSC [33]	Ocean floor	ROV	Clear	RGB	123	Points
The Brackish Dataset (Proposed)	Brackish strait	Stationary	Varying	RGB	14,518	Bounding box

Table D.1: An overview of annotated underwater image datasets.

3 Camera Setup

The setup used to capture the proposed dataset consists of three cameras and three lights. The devices are placed in a grid-wise manner on a stainless steel frame as illustrated in Figure D.2. However, the position and orientation of the devices in the figure are not representative for the arrangement used to capture the dataset.

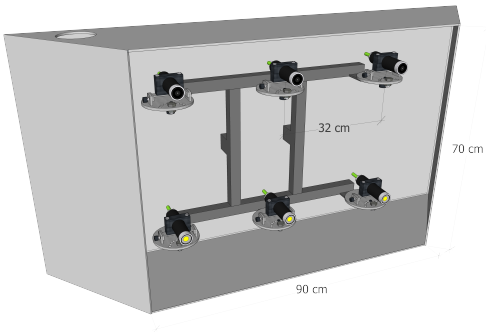


Fig. D.2: The setup consists of a stainless steel frame with three cameras and three lights.

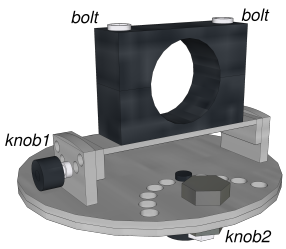


Fig. D.3: Adjustable mount for cameras and lights.

The cameras use a 1/3" Sony ExView Super HAD Color CCD imaging sensor and a 2.8 mm lens with a resolution up to 1080×1920 pixels and a framerate up to 30 fps with H.264 compression. The lamps are LEDs emitting light with 1900 lumens. Each camera and light is fitted in a cylindrical waterproof casing, which can resist a water pressure of approximately 10 bar. The diameter of the casing is 30 mm and the length is 128 mm.

As both the cameras and lights are placed in the same type of waterproof casing the setup is easily configurable, since all six positions in the steel frame can hold either lights or cameras. The design of the steel frame allows divers to adjust the orientation of lights and cameras under water. This is illustrated in Figure D.3, where *knob1* can be pulled in order to adjust the device vertically and *knob2*, hidden beneath the mount, can be pulled to adjust it horizontally. The two bolts on top of the mount can be loosened in order to change or replace a device.

The setup is permanently mounted on a pillar of the Limfjords-bridge connecting Aalborg and Nørresundby in Denmark, at a depth of around nine meters. A barrier of boulders are placed around the pillar for protection, but it also functions as a habitat for various marine species. The setup is therefore placed above this barrier as illustrated in Figure D.4, which is not to scale. The barrier is 6 meters high and slopes down towards the fairway between the pillars.

All cameras and lights are connected with cables which provides power and connects the devices to a Digital Video Recorder (DVR) system placed on the bridge. The DVR system is connected to the Internet, allowing for remote real time control and streaming from the cameras.

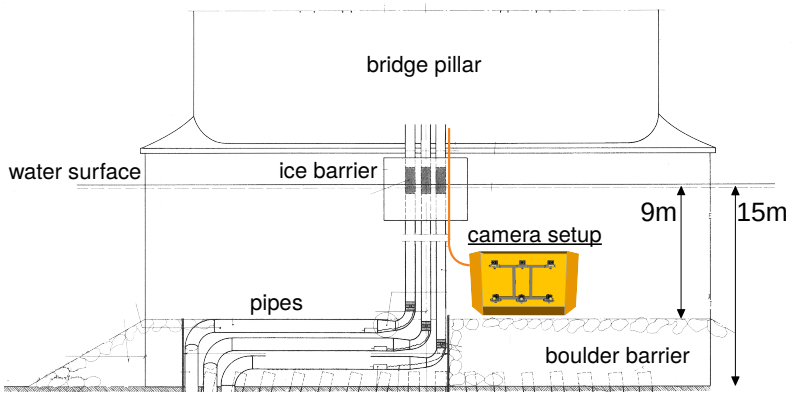


Fig. D.4: Drawing of the bridge pillar, boulder barrier, and the placement of the camera setup. The drawing is not to scale.

4 Dataset

Video data from the three cameras have been recorded since February 2019, currently resulting in more than 4,000 hours of video data. As the turbidity of the water and the activity from marine animals vary to a large degree, it is only a fraction of the recordings that is of interest seen from a computer vision perspective. A varied subset of 89 video clips captured between February 20 and March 25 has therefore been handpicked to be the foundation of the proposed dataset. All the chosen clips were captured from a single camera placed in the center position in the bottom of the frame, pointing towards the seafloor. One light, located directly to the left of the camera, seen from the camera's point of view, has been turned on at all times. The light is oriented towards the seafloor, but away from the camera's field of view in order to reduce backscatter.

The videos were categorized based on the main activity of the respective video and subsequently manually annotated with a bounding box annotation tool [37] resulting in a total of 14,518 frames with 25,613 annotations. The distribution of the annotations can be seen in Table D.2 where the number of annotations and amount of videos, where each class occur, are presented. It should be noted that multiple classes can occur in the same video.

Class	Annotations	Video Occurrences
Big fish	3,241	30
Crab	6,538	29
Jellyfish	637	12
Shrimp	548	8
Small fish	9,556	26
Starfish	5,093	30

Table D.2: Overview of the share of annotations and amount of video occurrences for the six respective classes.

Professor Niels Madsen, who is a marine biologist with expert knowledge on the local marine environment, has inspected the videos in order to help identify the various types of marine animals. However, due to the turbid recordings, and the relatively similar visual appearance of multiple fish species, it is extremely difficult to determine the exact species. The fish have therefore been coarsely classified as being either *big fish* or *small fish*.

The objects tagged as *big fish* are in most cases lumpsuckers (*Cyclopterus lumpus*) and can be seen in a gray/green or reddish variant depending on whether it is a male or female. However, the sculpin (*Myoxocephalus scorpius*) also visits the site and it can be difficult to tell the difference between the two when the water is turbid.

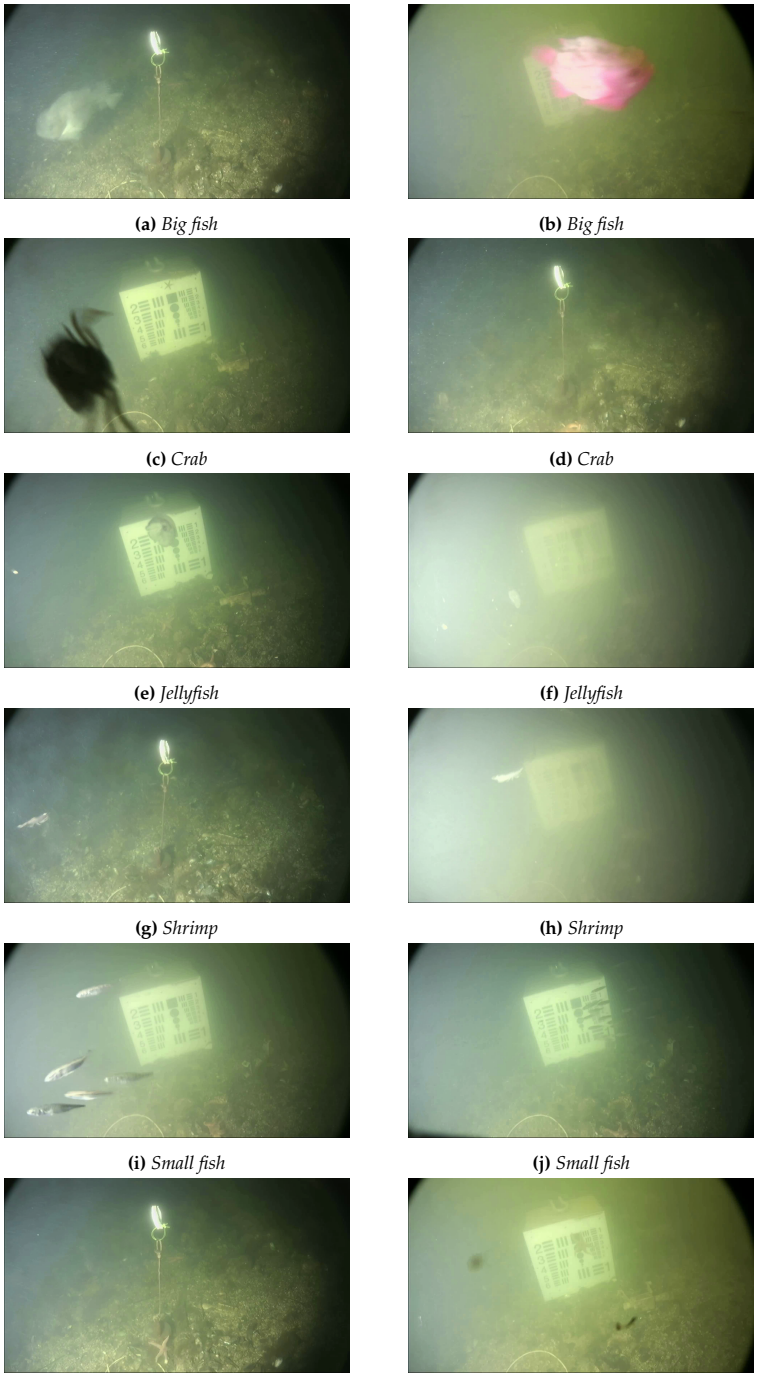
The *small fish* are in most cases sticklebacks (*Gasterosteus aculeatus*) when schools of fish appear in front of the camera. Other fish that have been observed include gobies (*Pomatoschistus*), European sprat (*Sprattus sprattus*), herrings (*Clupea harengus*), and eelpouts (*Zoarces viviparus*).

Most images contain various sizes of particles in the water, from dissolved matter to floating leaves and seaweed. These objects are not of immediate interest and are considered as noise in the images.

An object has been placed in the scene in front of the camera for other research purposes. For the first couple of weeks a floating device is visible (seen in Figure D.5a), while the videos from the last weeks contain a concrete block with a visible test pattern of size 40x40cm (seen in Figure D.5c).

Example frames with all object classes of the Brackish Dataset can be seen in Figure D.5. The images also show the variation of turbidity between the videos, e.g., Figure D.5h shows high turbidity, while Figure D.5c has low turbidity.

Limfjorden, where the videos have been recorded, is a 180 km long strait located between The North Sea and Kattegat. At the recording location the strait is approximately 500 m wide, up to 15 m deep, and at a distance of approximately 20 km to Kattegat. The water in the strait is brackish, which is a mixture between saltwater from the seas and freshwater from streams which ends up in the strait. The strength of the currents and winds in the strait can become relatively high and stir up sediments, which increases the turbidity. On other occasions, especially during summer, the water can be calm, resulting in a layered split between the heavy saline sea water and the lighter fresh water on top. The mean water temperature per month measured 1 meter below the surface can vary from 0.5°C to 18 °C.



(a) *Big fish* **(b)** *Big fish*

(c) *Crab* **(d)** *Crab*

(e) *Jellyfish* **(f)** *Jellyfish*

(g) *Shrimp* **(h)** *Shrimp*

(i) *Small fish* **(j)** *Small fish*

(k) *Starfish* **(l)** *Starfish*

Fig. D.5: Example-frames from the dataset, which illustrate the large in-class variation as well as the variation in turbidity.

5 Evaluation

Two state-of-the-art CNN based object detectors (YOLOv2 and YOLOv3 [38, 39]) are tested on the new dataset in order to obtain baseline results for future reference. Both networks have been fine-tuned in the Darknet framework [40]. YOLO is a convolutional neural network based single-shot object detector, which divides each image into regions and predicts bounding boxes and corresponding probabilities for each region. The probability, as a measure of confidence for a detection of a certain class, is used for weighting of each bounding box. As a single-shot object detector, YOLO processes the entire image and predicts all relevant bounding boxes in a single pass through the network, allowing for a high image per second processing rate.

5.1 Training

A brief explanation of the differences between the two pre-trained object detectors are:

- The YOLOv2 detector is obtained from the VIAME toolkit [41] and is pre-trained on ImageNet and fine-tuned on fish datasets from NOAA Fisheries Strategic Initiative on Automated Image Analysis.
- The YOLOv3 detector is used in its original version pre-trained on the Open Images dataset [39].

The YOLOv2 detector is already fine-tuned on underwater images, but contains only the two classes: *Vertebrates* and *Invertebrates*. Therefore, further fine-tuning is needed in order to be able to evaluate this model on the proposed dataset.

The YOLOv3 detector is trained on the Open Images dataset, which contains 601 classes where five of those are relevant: *fish*, *starfish*, *jellyfish*, *shrimp*, and *crab*.

Both models have subsequently been fine-tuned on the proposed dataset, described in Section 4, and tested with both Open Images and Brackish dataset categories on the proposed dataset. It should be noted that the only difference between the two categories is that the *fish* class in Open Images contains both the *big fish* and *small fish* from the Brackish categories.

The dataset is split randomly into 80 % training, 10 % validation and 10 % test data. Each network is trained for 30,000 iterations using their original training regime, only adjusting the batch size and setting the input size to 416×416, and with the earliest layer weights frozen.

5.2 Results

Object detectors are commonly evaluated based on the mean Average Precision (mAP) metric, which is the average precision calculated per category, averaged over all categories. The prediction bounding boxes are filtered by their Intersection over Union (IoU) with the ground truth bounding boxes. The MS COCO dataset [42] is the front running dataset used for object detection, segmentation, and more, which evaluates the mAP under different conditions.

The primary metric is the *AP@[IoU = 0.5:0.95]*, which is referred to as *AP*. *AP* is calculated as the averaged mAP, where the mAP values are calculated with an IoU threshold of [0.5, 0.55, 0.6, ..., 0.90, 0.95]. A metric with the IoU threshold set to 0.5 is also calculated, denoted AP_{50}, which is the primary metric of another large object detection dataset, PASCAL VOC [43]. For all the metrics, only the top 100 most confident predictions per image are included.

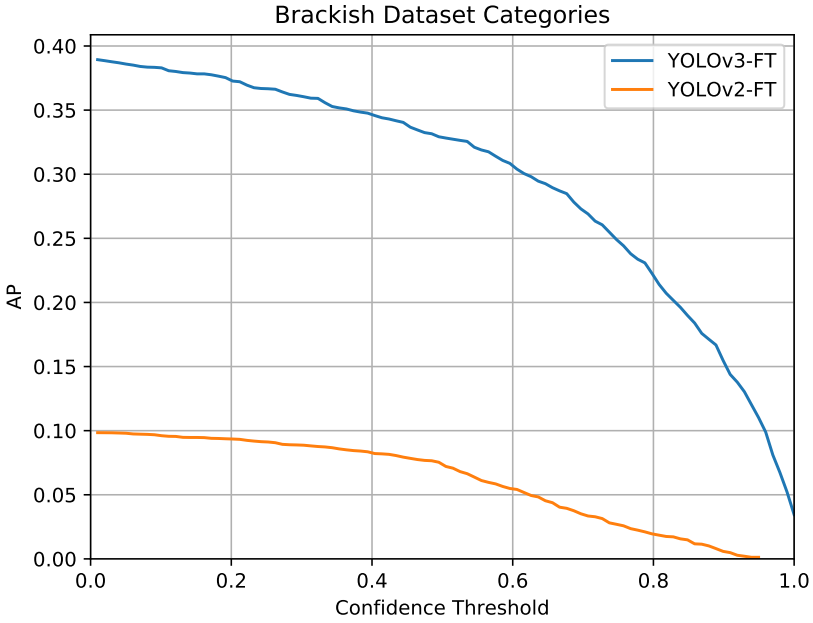


Fig. D.6: The *AP* metric as a function of the thresholded detection confidence, when using the Brackish dataset categories, for the fine-tuned models.

The *AP* is plotted against a prediction confidence threshold for the proposed dataset in Figure D.6. It can be seen that as the threshold is increased, the *AP* decreases. As the decrease is monotonic, it indicates that a large part of the correct predictions are with a low confidence. Therefore, a low confidence threshold of 0.01 is chosen when evaluating the trained networks.

The AP and the AP_{50}, which are the primary metrics of MS COCO and PASCAL VOC, are used as performance indicators. The networks have been evaluated on the new Brackish dataset with the Open Images categories, see Table D.3, and the Brackish categories, see Table D.4.

	Categories	AP	AP_{50}
YOLOv3	Open Images	0.0022	0.0035
YOLOv2 fine-tuned	Open Images	0.0748	0.2577
YOLOv3 fine-tuned	Open Images	**0.3947**	**0.8458**

Table D.3: Results of the models evaluated on the Open Images categories and compared by the AP and AP_{50} metrics.

	Categories	AP	AP_{50}
YOLOv2 fine-tuned	Brackish	0.0984	0.3110
YOLOv3 fine-tuned	Brackish	**0.3893**	**0.8372**

Table D.4: Results of the models evaluated on the Brackish categories and compared by the AP and AP_{50} metrics.

The results show that fine-tuning on the proposed dataset increases performance dramatically, but also that the achieved performance can still be improved.

Furthermore, a look into the per-class performance of the fine-tuned YOLOv3 network shows that the *starfish* category has a significantly higher score than the others. This is assumed to be due to both the distinctive shape and because the starfish rarely moves in the videos, leading to multiple annotations of starfish which are near identical.

The low score of *shrimp* and *small fish* is assumed to be due to their relatively small size and fast movement, which causes motion blur and loss of features.

Class	AP	AP_{50}
Big fish	0.4621	0.8999
Crab	0.4205	0.9271
Jellyfish	0.3746	0.8205
Shrimp	0.3238	0.7662
Small fish	0.2449	0.6229
Starfish	0.5102	0.9867

Table D.5: Per category results for the fine-tuned YOLOv3 model with the proposed Brackish dataset categories.

6 Future Work

The camera setup used to capture the proposed dataset has been developed in a way that allows for easy maintenance, adjustments, and replacement. It is a permanent setup that will be used to monitor the various species that visit the area during the seasons. At the moment the three cameras are pointing diagonally downwards toward the riverbed, but there will be made ongoing modifications in order to capture different types of dataset, including stereo sequences.

The proposed dataset is part of an ongoing research project where data is logged 24 hours a day from all three cameras. However, the recordings are stored in a compressed format in order to reduce the amount of data. In the future, the plan is to expand the current dataset with uncompressed recordings.

The largest species that has been observed on the recordings is the harbor seal (*Phoca vitulina*), which is a protected animal in national waters and of great interest for marine researchers. The seal is not a part of the proposed dataset due to the few encounters so far, but hopefully will be in the future as more data is gathered and specific species are added to the dataset in close collaboration with local marine biologists.

7 Conclusion

A new bounding box annotated image dataset of marine animals, recorded in brackish waters, is presented in this paper. The dataset consists of 14,518 frames with 25,613 annotations of the six classes: *big fish*, *small fish*, *crab*, *jellyfish*, *shrimp*, and *starfish*. To the best of knowledge, the proposed dataset is unique, as it is the only annotated image dataset captured in temperate brackish waters.

Two state-of-the-art CNNs (YOLOv2 and YOLOv3) has been fine-tuned on the proposed Brackish Dataset and evaluated in order to create a baseline for future reference. The YOLOv2 object was pre-trained on Imagenet and fine-tuned to fish datasets and it was obtained from the VIAME toolkit [41]. The YOLOv3 detector was the original version pre-trained on the Open Images dataset [39]. The evaluation is based on the primary metrics of the MS COCO and PASCAL VOC, which are both based on the mean Average Precision (mAP). The fine-tuned YOLOv3 network achieved the best performance with $AP \approx 39\%$ and $AP_{50} \approx 84\%$, allowing for improvements to be made.

The proposed Brackish Dataset has been made publicly available at `https://www.kaggle.com/aalborguniversity/brackish-dataset`

References

[1] E. Lindstrom, J. Gunn, A. Fischer, A. McCurdy, L. K. Glover, and T. T. Members, "A framework for ocean observing," Tech. Rep., 2012.

[2] N. G. Jerlov, *Marine Optics*. Elsevier, 1976, vol. 14.

[3] C. D. Mobley, *Light and water: radiative transfer in natural waters*. Academic press, 1994.

[4] A. Abramic, D. Gonzalez, E. Bigagli, A. Che-Bohnenstengel, and P. Smits, "INSPIRE: Support for and requirement of the marine strategy framework directive," *Marine Policy*, vol. 92, pp. 86–100, 2018.

[5] A. Crise, M. R. d'Alcalà, P. Mariani, G. Petihakis, J. Robidart, D. Iudicone, R. Bachmayer, and F. Malfatti, "A conceptual framework for developing the next generation of marine OBservatories (MOBs) for science and society," *Frontiers in Marine Science*, vol. 5, 2018.

[6] S. Villon, M. Chaumont, G. Subsol, S. Villéger, T. Claverie, and D. Mouillot, "Coral reef fish detection and recognition in underwater videos by supervised machine learning: Comparison between deep learning and hog+svm methods," in *Advanced Concepts for Intelligent Vision Systems*. Cham: Springer International Publishing, 2016, pp. 160–171.

[7] A. Salman, A. Jalal, F. Shafait, A. Mian, M. Shortis, J. Seager, and E. Harvey, "Fish species classification in unconstrained underwater environments based on deep learning," *Limnology and Oceanography: Methods*, vol. 14, no. 9, pp. 570–585, 2016.

[8] A. Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, W.-P. Vellinga, R. Planqué, A. Rauber, R. Fisher, and H. Müller, "Lifeclef 2014: Multimedia life species identification challenges," in *Information Access Evaluation. Multilinguality, Multimodality, and Interaction*. Cham: Springer International Publishing, 2014, pp. 229–249.

[9] A. Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, W.-P. Vellinga, R. Planqué, A. Rauber, S. Palazzo, B. Fisher, and H. Müller, "Lifeclef 2015: Multimedia life species identification challenges," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*. Cham: Springer International Publishing, 2015, pp. 462–483.

[10] S. A. Siddiqui, A. Salman, M. I. Malik, F. Shafait, A. Mian, M. R. Shortis, and E. S. H. and, "Automatic fish species classification in underwater videos: exploiting pre-trained deep neural network models to compensate for limited labelled data," *ICES Journal of Marine Science*, vol. 75, no. 1, pp. 374–389, 2017.

[11] M. Dawkins, C. Stewart, S. Gallager, and A. York, "Automatic scallop detection in benthic environments," in *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE, 2013.

[12] C. Rasmussen, J. Zhao, D. Ferraro, and A. Trembanis, "Deep census: AUV-based scallop population monitoring," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*. IEEE, 2017.

[13] A. Mahmood, M. Bennamoun, S. An, F. A. Sohel, F. Boussaid, R. Hovey, G. A. Kendrick, and R. B. Fisher, "Automatic annotation of coral reefs using deep learning," in *OCEANS 2016 MTS/IEEE Monterey*. IEEE, 2016.

[14] M. Bewley, A. Friedman, R. Ferrari, N. Hill, R. Hovey, N. Barrett, O. Pizarro, W. Figueira, L. Meyer, R. Babcock, L. Bellchambers, M. Byrne, and S. B. Williams, "Australian sea-floor survey data, with images and expert annotations," *Scientific Data*, vol. 2, no. 1, p. 150057, oct 2015.

[15] O. Beijbom, T. Treibitz, D. I. Kline, G. Eyal, A. Khen, B. Neal, Y. Loya, B. G. Mitchell, and D. Kriegman, "Improving automated annotation of benthic survey images using wide-band fluorescence," *Scientific Reports*, vol. 6, no. 1, 2016.

[16] R. B. Fisher, Y.-H. Chen-Burger, D. Giordano, L. Hardman, and F.-P. Lin, *Fish4Knowledge: Collecting and Analyzing Massive Coral Reef Fish Video Data*. Springer International Publishing, 2016, vol. 104.

[17] A. Salman, S. A. Siddiqui, F. Shafait, A. Mian, M. R. Shortis, K. Khurshid, A. Ulges, and U. Schwanecke, "Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system," *ICES Journal of Marine Science*, 2019.

[18] S. Hasija, M. J. Buragohain, and S. Indu, "Fish species classification using graph embedding discriminant analysis," in *Proceedings of the International Conference on Machine Vision and Information Technology (CMVIT)*. IEEE, 2017.

[19] X. Li, M. Shang, J. Hao, and Z. Yang, "Accelerating fish detection and recognition by sharing CNNs with objectness learning," in *OCEANS 2016 - Shanghai*. IEEE, 2016.

[20] Z. Cao, J. C. Principe, B. Ouyang, F. Dalgleish, and A. Vuorenkoski, "Marine animal classification using combined CNN and hand-designed image features," in *OCEANS 2015 - MTS/IEEE Washington*. IEEE, 2015.

[21] A. Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, W.-P. Vellinga, J. Champ, R. Planqué, S. Palazzo, and H. Müller, "Lifeclef 2016: Multimedia life species identification challenges," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*. Cham: Springer International Publishing, 2016, pp. 286–310.

[22] J. A. for Marine-Earth Science and Technology, "JAMSTEC E-library of deep-sea images," 2016, http://www.godac.jamstec.go.jp/jedi/e/.

[23] H. Lu, Y. Li, T. Uemura, Z. Ge, X. Xu, L. He, S. Serikawa, and H. Kim, "FDCNet: filtering deep convolutional network for marine organism classification," *Multimedia Tools and Applications*, vol. 77, no. 17, pp. 21 847–21 860, 2017.

[24] Y. Li, H. Lu, J. Li, X. Li, Y. Li, and S. Serikawa, "Underwater image de-scattering and classification by deep neural network," *Computers & Electrical Engineering*, vol. 54, pp. 68–77, 2016.

[25] J. Jäger, M. Simon, J. Denzler, V. Wolff, K. Fricke-Neuderth, and C. Kruschel, "Croatian fish dataset: Fine-grained classification of fish species in their natural habitat," in *Procedings of the Machine Vision of Animals and their Behaviour Workshop 2015*. British Machine Vision Association, 2015.

[26] K. Anantharajah, Z. Ge, C. McCool, S. Denman, C. Fookes, P. Corke, D. Tjondronegoro, and S. Sridharan, "Local inter-session variability modelling for object classification," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2014.

[27] R. Taylor, N. Vine, A. York, S. Lerner, D. Hart, J. Howland, L. Prasad, L. Mayer, and S. Gallager, "Evolution of a benthic imaging system from a towed camera to an automated habitat characterization system," in *OCEANS 2008*. IEEE, 2008.

[28] N. F. S. Center, "Habitat mapping camera (HABCAM)," 2017, https://inport.nmfs.noaa.gov/inport/item/27598.

[29] A. C. for Field Robotics, "Tasmania coral point count," http://marine.acfr.usyd.edu.au/datasets/.

[30] M. Bewley, B. Douillard, N. Nourani-Vatani, A. Friedman, O. Pizarro, and S. B. Williams, "Automated species detection: An experimental approach to kelp detection from sea-floor auv images," in *Proceedings of Australasian Conference on Robotics and Automation*, 2012.

[31] O. Beijbom, P. J. Edmunds, D. I. Kline, B. G. Mitchell, and D. Kriegman, "Automated annotation of coral reef survey images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012.

[32] L. B. Preskitt, P. S. Vroom, and C. M. Smith, "A rapid ecological assessment (REA) quantitative survey method for benthic algae using photoquadrats with scuba," *Pacific Science*, vol. 58, no. 2, pp. 201–209, 2004.

[33] C. . Workshop and C. A. A. of Marine Video for Environmental Monitoring, "Data challenge description," 2018, http://www.viametoolkit.org/cvpr-2018-workshop-data-challenge/challenge-data-description/.

[34] I. Kavasidis, S. Palazzo, R. D. Salvo, D. Giordano, and C. Spampinato, "An innovative web-based collaborative platform for video annotation," *Multimedia Tools and Applications*, vol. 70, no. 1, pp. 413–432, 2013.

[35] B. J. Boom, P. X. Huang, J. He, and R. B. Fisher, "Supporting ground-truth annotation of image datasets using clustering," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, 2012.

[36] C. Beyan and R. B. Fisher, "Detecting abnormal fish trajectories using clustered and labeled data," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2013.

[37] C. H. Bahnsen, A. Møgelmose, and T. B. Moeslund, "The aau multimodal annotation toolboxes: Annotating objects in images and videos," *arXiv preprint arXiv:1809.03171*, 2018.

[38] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," *arXiv:1612.08242*, 2016.

[39] ——, "Yolov3: An incremental improvement," *arXiv:1804.02767*, 2018.

[40] J. Redmon, "Darknet: Open source neural networks in c," http://pjreddie.com/darknet/, 2013–2016.

[41] M. Dawkins, L. Sherrill, K. Fieldhouse, A. Hoogs, B. Richards, D. Zhang, L. Prasad, K. Williams, N. Lauffenburger, and G. Wang, "An open-source platform for underwater image and video analytics," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017.

[42] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Computer Vision – ECCV 2014*. Cham: Springer, 2014, pp. 740–755.

[43] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision (IJCV)*, vol. 88, no. 2, pp. 303–338, 2010.

Paper E

BrackishMOT: The Brackish Multi-Object Tracking Dataset

Malte Pedersen, Daniel Lehotský,
Ivan Nikolov, and Thomas B. Moeslund

The paper has been published in
Image Analysis. SCIA 2023, LNCS 13885, pp. 17-33.

The layout has been revised.

Abstract

There exist no publicly available annotated underwater multi-object tracking (MOT) datasets captured in turbid environments. To remedy this we propose the BrackishMOT dataset with focus on tracking schools of small fish, which is a notoriously difficult MOT task. BrackishMOT consists of 98 sequences captured in the wild. Alongside the novel dataset, we present baseline results by training a state-of-the-art tracker. Additionally, we propose a framework for creating synthetic sequences in order to expand the dataset. The framework consists of animated fish models and realistic underwater environments. We analyse the effects of including synthetic data during training and show that a combination of real and synthetic underwater training data can enhance tracking performance. Links to code and data can be found at `https://www.vap.aau.dk/brackishmot`.

1 Introduction

Humans have relied on the oceans as a steady food-source for millennia, but the marine ecosystems are now rapidly deteriorating due to human impact. This is especially a problem for coastal societies across the globe that rely on fish as their main food-source and for biodiversity in general. The severity is underlined by the fact that the United Nations included *Life Below Water* as the fourteenth Sustainable Development Goal (UN SDG #14) [1]. The increase in attention to monitoring the condition of the ocean has entailed pressure on marine researchers to gather data at an unprecedented pace. However, traditional marine data-gathering methods are often time-consuming, intrusive, and difficult to scale as they require organisms to be caught and measured manually. Therefore, it is critical that assistive solutions for optimizing and scaling data-gathering in marine environments are developed.

During the past decade, computer vision solutions have increased dramatically in performance due to the utilization of strong graphics processing units (GPU) combined with the popularisation of deep learning algorithms. Simultaneously, underwater cameras have become significantly better and cheaper [2]. This calls for marine researchers to utilize both cameras and computer vision to scale data-gathering right away, however, this is currently not feasible as there is a critical lack of marine datasets for training and evaluating marine computer vision models.

To remedy this gap we propose a novel multi-object tracking (MOT) dataset of marine organisms in the wild named BrackishMOT. In short, MOT describes the task of obtaining the trajectories of all objects in the scene. The trajectories are obtained by having a model detect objects spatially and associating the detections temporally. Tracking is a core component in marine research and can be used for multiple purposes such as counting or conducting behavioral

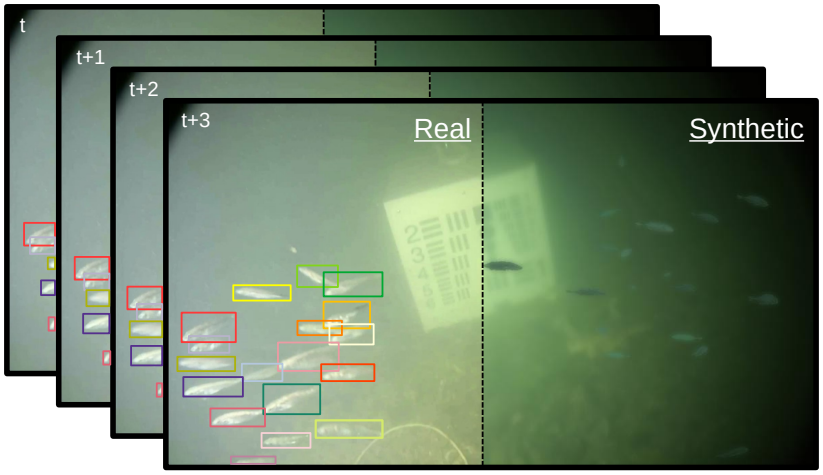


Fig. E.1: We present and publish a bounding box annotated underwater multi-object tracking dataset captured in the wild named BrackishMOT, together with a synthetic framework for generating more data for which we publish both data and source code.

analysis.

Manually annotated datasets are critical and necessary for evaluating the performance of trackers on data from the wild. However, they are not scalable and do not necessarily generalize well to environments that are not included in the dataset. Therefore, we investigate how synthetic underwater sequences can be used for training multi-object trackers. We develop a framework for creating synthetic sequences that resemble the BrackishMOT environment and analyse how key factors, namely turbidity, floating particles, and the background, affect tracker performance. This is a critical step toward the development of new high-quality underwater synthetic datasets. Our contributions are summarized below:

Contributions

- We present and publish BrackishMOT, a novel MOT dataset captured in brackish waters with a total of 98 sequences and six different classes.
- We propose a framework for creating synthetic underwater sequences based on phenomena observed in the wild and analyse their effect on tracker performance.
- We analyse different training strategies for a state-of-the-art tracker using both real and synthetic data and present baseline results for BrackishMOT.

The current state-of-the-art MOT algorithms like Tracktor [3], CenterTrack [4], FairMOT [5], and ByteTrack [6] have all been developed for tracking terrestrial objects like pedestrians and vehicles. Common denominators for these types of objects are relatively predictable motion and typically strong visual cues. However, in the underwater domain, most objects like fish are prey animals which means that they may behave erratically to avoid being tracked by predators. Furthermore, objects of the same species often look very similar and provide weak visual features for re-identification. In other words, the trackers need to be re-trained or fine-tuned to cope with the challenges of the underwater domain. Modern trackers are generally based on deep learning and require large amounts of training data. While there exist multiple terrestrial tracking datasets like KITTI [7], MOTChallenge [8–10], and UAVDT [11], there are only few publicly available datasets with underwater objects. In this section, we dive into the sparse literature on underwater datasets and trackers.

1.1 Underwater MOT Datasets

Compared to its terrestrial counterpart, the underwater MOT domain has not witnessed a noteworthy increase in novel algorithms during the past decade. One of the reasons for this lack of algorithms dedicated to the underwater domain is the low number of publicly available annotated underwater datasets suitable for training and evaluating modern algorithms.

Underwater MOT datasets can generally be split into two categories: controlled environments such as aquariums [12, 13], and the more challenging uncontrolled natural underwater environments which we will focus on in this paper. One of the earliest datasets used for tracking of fish captured in the wild was the Fish4Knowledge (F4K) dataset [14, 15]. The F4K dataset was captured more than a decade ago in mostly clear tropical waters off the coast of Taiwan in very low resolution and low frame rate. More recently, the two underwater object tracking datasets UOT32 [16] and UOT100 [17] were published with annotated underwater sequences sourced from YouTube videos. The UOT32 and UOT100 datasets provide sequences from diverse underwater environments but are focused on single object tracking. Lastly, a high-resolution underwater MOT dataset captured off the coast of Hawai'i island named FISHTRAC [18] was recently proposed. However, at the time of writing only three training videos (671 frames in total) with few objects and little occlusion have been published.

The datasets captured in tropical waters only cover a tiny fraction of the diverse underwater ecosystems. The conditions in many other areas are far less favorable with less colorful fish and more turbid water. To advance the research in underwater MOT, it is critical to developing new datasets captured in other and more challenging environments and we see the BrackishMOT dataset as an important contribution to this field.

1.2 Underwater Trackers

Relatively few multi-object trackers dedicated to the underwater domain exist with most of them developed for tracking fish in controlled environments [12, 13, 19, 20]. A common trait for trackers developed for controlled environments is the assumption of good detections and strong visual cues for re-identification. This is generally not the case in uncontrolled environments, where the light may change, the water is turbid, the background varies, and algae may bloom on the lens [14, 21].

To tackle these problems the team behind the F4K dataset proposed a method for detecting fish using mixture models for background subtraction and handcrafted features based on motion and color for classifying fish from other objects [14]. Lastly, they modeled every track by feature-based covariance matrices based on representations from previous frames and associated new detections by minimizing the distance between the covariance matrices. Another group that also worked on the F4K tracking data experimented with AlexNet [22] and VGG-19 [23] as feature extractors for appearance-based association and used a directed acyclic graph in a two-step approach by first constructing strong local tracklets followed by a tracklet-association step for finalizing the tracks [24].

Recently, a few groups have proposed trackers evaluated on new annotated underwater datasets. Liu et al. proposed a multi-class tracker named RMFC [25] utilizing YOLOv4 [26] as the backbone for a detection and tracking branch running in parallel, which showed promising results. In the work by Martija et al. [27] they investigated the use of synthetic data to enhance tracker performance. They propose to use Faster R-CNN [28] for object detection and a deep hungarian network [29] for associating detections temporally using visual cues. Unfortunately, both groups evaluate their method on private datasets, and they have not shared their code.

The most recent work on fish tracking in the wild is the work done by Mandel et al. [18]. They propose an offline tracker utilizing a greedy approach that initializes a track from the strongest detection across all frames based on a confidence score. Detections in previous and future frames are associated with the track based on appearance and motion. When a track has been finalized, the next track is built in the same manner, and so forth. They evaluated their tracker with detections from YOLOv4 and RetinaNet [30].

Common for the aforementioned methods is a reliance on strong visual cues or predictable motion for associating detections. This works well for scenes with few objects, in clear tropical waters, or if the objects are visually distinct. This is a natural consequence caused by the limited datasets used in the development of the methods and it exposes the need for diverse datasets to represent the variety of underwater ecosystems.

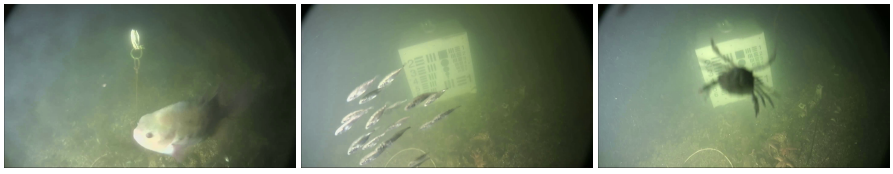
1.3 Synthetic Underwater Data

A way to remedy the scarce amount of publicly available underwater datasets is to use synthetic data. A typical approach to produce synthetic underwater data (in 2D) is by pasting cutouts of the organisms onto some background. Mahmood et al. [31] used this method to place manually segmented parts of lobsters, like the body or the antennas, onto a diverse set of backgrounds sampled from the Benthoz15 [32] dataset to generate synthetic data suitable for lobster detection in heavily occluded scenes. Martija et al. [27] used weakly generated bounding boxes and masks to simulate the movement of fish across a background to create rough synthetic MOT data. And for developing an underwater litter detector Music et al. [33] pasted various 3D shapes into real-life underwater images to create training data.

An alternative to the 2D approach is data generation from animated 3D scenes. In 1987, Reynolds proposed the boid model for accurately simulating the behaviour of fish schools [34]. The boid behaviour model has since been extended multiple times [35–37], with [38] combining their variation of boids with a synthetic data generator to produce realistic annotated underwater data from animated sequences of fish schools. However, to the best of our knowledge, there has been no attempt to produce synthetic underwater MOT data based on boid behavior. In the next section, we will introduce and describe our new underwater MOT dataset followed by a description of our framework for creating synthetic underwater sequences.

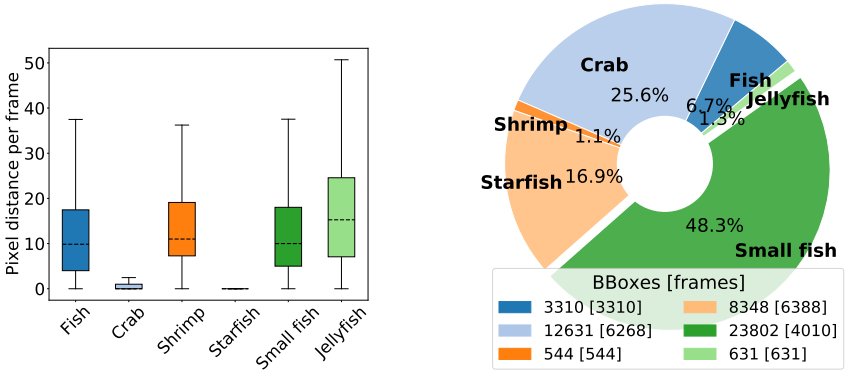
2 The BrackishMOT Dataset

In 2019 the Brackish Dataset [39] was published. Its purpose was to advance object detection in brackish waters. It has been popular in the community since it was the first underwater detection dataset captured in non-tropical waters. The recordings of the dataset were captured nine meters below the surface in brackish waters and consist of 89 sequences in total. The sequences contain manually annotated bounding boxes of six coarse classes: *fish*, *crab*,



(a) Fish **(b)** Small fish **(c)** Crab

Fig. E.2: Image samples from the Brackish Dataset [39]. In a majority of the sequences containing the *small fish* class, there are multiple specimens forming a school of fish.



(a) Boxplot showing the distribution of the traveled distance between consecutive frames measured in pixels.

(b) Class distribution of the brackishMOT dataset based on the number of bounding boxes.

Fig. E.3: Plots describing the composition of the brackishMOT dataset with respect to motion and class distribution. For both plots, the data is from all the sequences.

shrimp, *starfish*, *small fish*, and *jellyfish*. Examples from the original dataset can be seen in Figure E.2.

2.1 Dataset Overview

In this work, we propose to expand the Brackish Dataset to include a MOT task. Therefore, we provide a new set of ground truth annotations for every sequence, based on the MOTChallenge annotation style [10]. Additionally, we present 9 new sequences focused on the *small fish* class, which gives a total of 98 sequences for the MOT task of the Brackish Dataset which we name **BrackishMOT**. The *small fish* class is especially relevant for the MOT task as it contains species that exhibit social and schooling behavior as illustrated in Figure E.2b. The ground truth files are comma-separated and include annotations per object in the following structure:

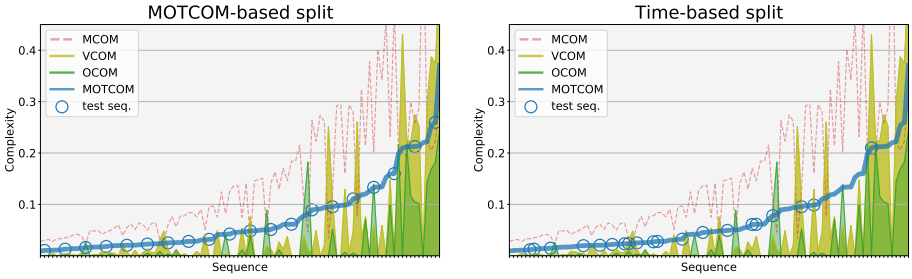
```
<frame>, <id>, <left>, <top>, <width>, <height>,
<confidence>, <class>, <visibility>
```

where `left` and `top` are the x and y coordinates of the object's top-left corner of the bounding box. Together with the `width` and `height` they describe the object's bounding box in pixels. `confidence` and `visibility` are both set to 1 and an object keeps its `id` as long as it is within field of view. There are rare cases where objects gets fully occluded in-frame and it is ambiguous to decide the ID of the object as it re-appears; in these cases, the object acquires a new ID. The `class` is in the range 1-6 where: (1) *fish*, (2) *crab*, (3) *shrimp*, (4) *starfish*, (5) *small fish*, and (6) *jellyfish*.

In Figure E.3 we present two charts illustrating the motion and class distribution for the dataset. We see that the *crab* and *starfish* classes barely move compared to the rest. In addition to that, they are well-camouflaged and most often move along the seabed. This constitutes a specific task as they are both hard to detect visually and from motion cues. The class distribution presented in Figure E.3b shows that the dataset is imbalanced with few occurrences of the *shrimp*, *fish*, and *jellyfish* classes. Furthermore, as the number of bounding boxes and frame occurrences are equal we can decipher that these three classes occur in the sequences as single objects. As the *small fish* class is the only class that exhibits erratic motion and appears in groups it is deemed the most interesting class with respect to MOT.

2.2 BrackishMOT Splits

Creating balanced training and testing splits is important to ensure a fair evaluation of the tracker performance and to give an accurate depiction of the task. To a large degree, this is a problem that has been overlooked in the creation of most MOT datasets due to the lack of a suitable metric. This has changed with the recent introduction of the MOTCOM framework [40]. MOTCOM is a metric that can estimate the complexity of MOT sequences based on the ground truth annotations and lay the foundation for creating more balanced data splits. The metric is a combination of three sub-metrics that describe the level of occlusion (OCOM), non-linear motion (MCOM), and visual similarity (VCOM) for every sequence. MOTCOM and the sub-metrics are all in the interval from 0 to 1 where a higher MOTCOM score means a more complex problem. We aim to create splits that are approximately evenly complex.



(a) Sorting the sequences based on MOTCOM and taking every fifth to be included in the test split. This is the approach we follow.

(b) A typical test split consisting of the 20 first recorded sequences. This approach clearly skews the splits with respect to complexity.

Fig. E.4: These plots illustrate MOTCOM and the sub-metrics for all the BrackishMOT sequences. In both plots, the sequences are sorted based on their MOTCOM score. The circles mark the test sequences with respect to the split-scheme.

In Figure E.4a we present MOTCOM and the sub-metrics for each sequence of the BrackishMOT dataset. The metrics are calculated on basis of all six classes combined. We see that the motion varies a lot between the individual sequences. However, even though the motion is quite non-linear and complex for several sequences then both occlusion and visual similarity are very low. This is due to the generally low number of objects in the scenes. A single jellyfish or shrimp may move fast and non-linearly, but if they are alone or in a scene with just a few objects they are less likely to be occluded or confused with other individuals.

The sequences containing the *small fish* class are generally exceptions to the above as they tend to score higher values in all three sub-metrics compared to the other sequences. These sequences often include fish schools which means that they have a higher number of objects that moves more around and are more social compared to e.g., starfish and crabs on the seabed. Therefore, the objects are more likely to be occluded and they are easier to confuse with each other as they look visually similar.

We create the splits based on the following scheme: we sort the sequences according to their MOTCOM score, then we pick the sequence with the highest MOTCOM score to be in the train split, the second highest goes into the test split, and from then on every fifth sequence goes into the test split while the rest goes into the train split. This gives a total of 20 test sequences illustrated by the circles in Figure E.4a and 78 train sequences. If we, on the other hand, had chosen a typical scheme like picking the first 20 recorded sequences to be included in the test split and the rest in the train split, we would have had a significantly different dataset structure as illustrated in Figure E.4b. With such a composition it is likely that trackers would generally perform better when evaluated on the test split, but it would be on false terms as the train split is significantly more challenging compared to the test split. The opposite can of course also happen, but that is equally problematic. For this reason, we make an informed split based on the MOTCOM scores.

To extend the proposed BrackishMOT dataset, we have developed a framework for creating synthetic underwater sequences based on phenomena observed in the BrackishMOT data as we believe that synthetic data is critical as a means to scale the availability of underwater training data. The framework is described in the next section.

3 Synthetic Data Framework

The proposed synthetic framework is built within the Unity game engine [41], using the built-in rendering pipeline and it is based on three main components: providing realistic fish meshes, modeling fish behavior, and building a realistically looking underwater environment. We provide options for each

(a) High-resolution mesh. **(b)** Low-resolution mesh with rig.

Fig. E.5: Illustration of the *stickleback* fish model used in our framework. (a) Initial high-resolution model and (b) decimated and rigged model for Unity.

of the components in order to create a synthetic environment that resembles the BrackishMOT data, however, the proposed framework is easily extendable with other species, behavior models, and surroundings. We will describe each of the components in the following sections.

Fish Model

An illustration of the fish model used in our framework can be seen in Figure E.5. The model was taken from the fish database of images and photogrammetry 3D reconstructions [42] and was selected as it visually resembles a *stickleback*, which is the family of the *small fish* class that most often occurs in schools in the Brackish dataset sequences according to the authors [39]. The 3D input model, shown in Figure E.5a, was decimated to 11,000 vertices. In order to preserve finer details of the mesh, a normal map was created from the high-resolution texture. The down-sampled mesh can be seen in Figure E.5b. Lastly, the model was rigged using the bones system in Blender [43] to allow for smooth animations of the body and tail.

The number of spawned fish in each sequence is randomly selected within a range between 4 and 50 to resemble the diversity of the BrackishMOT sequences. The initial pose, scale, and appearance (texture albedo and glossiness) for each fish varies between the sequences. A table with all the randomized parameters and their respective ranges can be found in the supplementary material.

Behavior Model

To approximate realistic fish schooling behavior, we use a boid-based behavioral model inspired by the work of C.W. Reynolds [34] and C. Hartman and B. Benes [36]. Each fish considers the position and heading of all other fish in its neighborhood. For each fish the velocity and heading is dependent on four factors: separation $\vec{s}$, cohesion $\vec{k}$, alignment $\vec{m}$, and leader $\vec{l}$. Separation

ensures avoidance of collisions with other members of the school. Cohesion is a force that drives the fish to seek the center of the neighborhood. Alignment is the drive of individual fish to match the others' velocity. Leader is a direction towards where a given leader is heading and for each fish, the leader is the neighbor with a heading vector closest to the fish's own heading vector. The steering vector is given by

$$\vec{steer} = S\vec{s} + K\vec{k} + M\vec{m} + L\vec{l}, \tag{E.1}$$

where S, K, M, and L are weights for the separation, cohesion, alignment, and leader forces, respectively. A more detailed description of the behavior model can be found in the supplementary material.

The Surrounding Environment

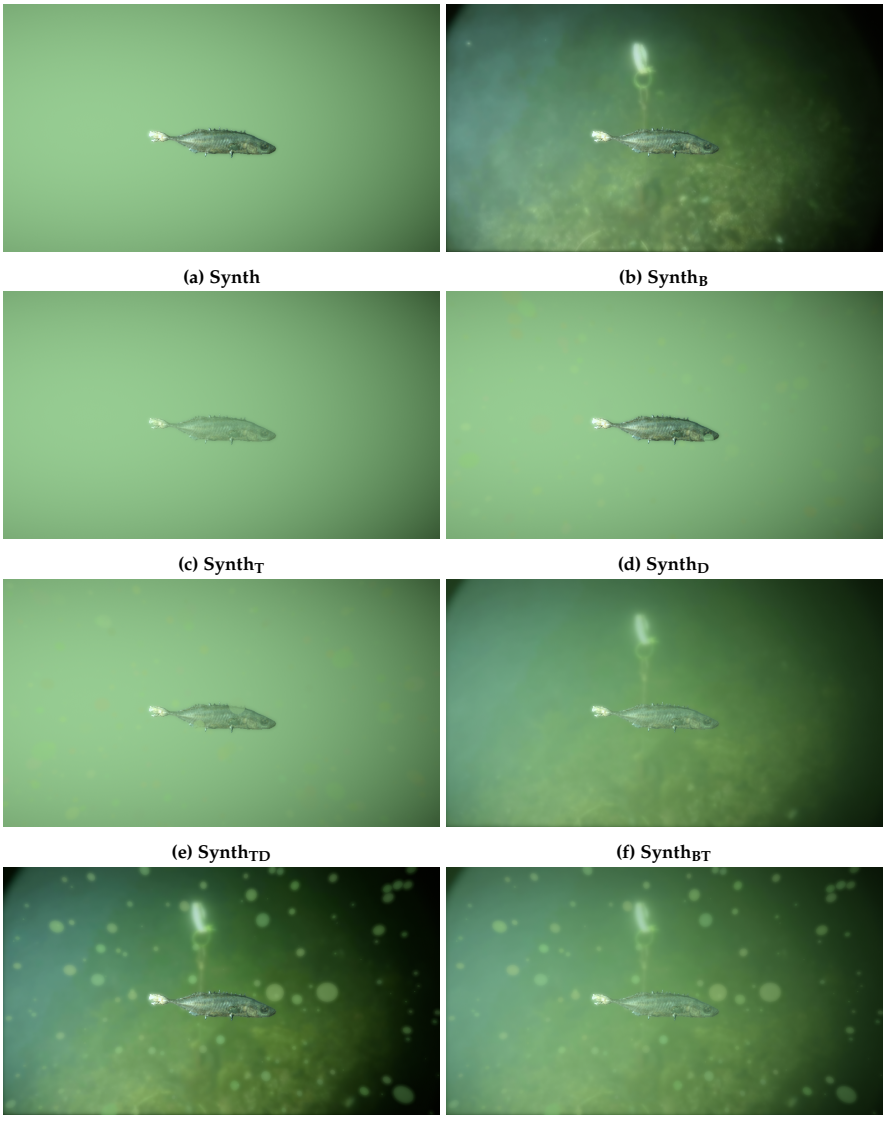
To investigate how changes in the environment impact tracker performance, we design the synthetic environment based on three variables: turbidity, background, and distractors.

Turbidity represents tiny floating particles in the water that engulfs the scene like a fog that intensifies as the distance between the object and the camera increases. The visibility varies to a large degree in the BrackishMOT sequences due to this phenomenon. We implement the turbidity effect using a custom-made Unity material with adjustable transparency and post-processing effects of depth of field and color grading. The color of the material spans between grey and green to resemble the turbidity observed in the BrackishMOT sequences. Both the color and intensity vary between the generated sequences.

We use videos from the Brackish dataset without fish as the background to make the scene more realistic. We include a range of background sequences and augment them by saturation, color, and blur to increase variation. When no background is present, we use a monotone color that matches the color of the turbidity.

Lastly, we introduce distractors [44, 45], which represent floating particles. The BrackishMOT sequences have been captured in shallow water where the current is often strong. The combination of strong current and shallow water induces floating plant material and resuspended sediments often occur in front of the camera as unclear circular bodies. To simulate this phenomenon we implement distractors as spheres with varying scales, levels of transparency, and color. The color range spans between grey and green as with the turbidity and monotone background. The number of distractors vary between the sequences and each distractor is spawned in a random position and is randomly moved to a new position between each frame.

Each of the environmental variables adds a layer of complexity to the synthetic scene based on phenomena observed in the real sequences. Com-



(a) Synth **(b) Synth$_{\mathbf{B}}$**

(c) Synth$_{\mathbf{T}}$ **(d) Synth$_{\mathbf{D}}$**

(e) Synth$_{\mathbf{TD}}$ **(f) Synth$_{\mathbf{BT}}$**

(g) Synth$_{\mathbf{BD}}$ **(h) Synth$_{\mathbf{BTD}}$**

Fig. E.6: Visualisation of different conditions of our synthetic environment. **(a)** Plain background, no turbidity, no distractors **(Synth)**. **(b)** Video background, no turbidity, no distractors **(Synth$_{\mathbf{B}}$)**. **(c)** Plain background, turbidity, no distractors **(Synth$_{\mathbf{T}}$)**. **(d)** Plain background, no turbidity, with distractors **(Synth$_{\mathbf{D}}$)**. **(e)** Plain background, with turbidityand distractors **(Synth$_{\mathbf{DT}}$)**. **(f)** Video background with turbidity, but without distractors **(Synth$_{\mathbf{BT}}$)**. **(g)** Video background with distractor, but no turbidity **(Synth$_{\mathbf{BD}}$)**. **(h)** Video background with turbidity and distractors **(Synth$_{\mathbf{BTD}}$)**.

binatorial variations of the variables give us eight synthetic environments, which can be seen in Figure E.6. Each generated video sequence is 10 seconds long and contains 150 frames animated with a frame rate of 15 FPS, which resembles the sequences of the BrackishMOT dataset. We include 50 sequences for each environment variation. The synthetic framework is general as all parameters can be adjusted to fit other underwater environments, e.g., using another video background would significantly alter the visuals of the sequences, or one could change the current or add new models to the scene. Source code and guides to using the framework can be found on the project page `https://vap.aau.dk/brackishmot/`.

4 Experiments

It is notoriously difficult for humans to visually track fish of the same species in video sequences captured in the wild. This is especially true when the water is turbid, the camera resolution is low, and the objects swim close to each other as is the case in some of the BrackishMOT sequences. This indicates that visual cues for re-identifying the objects are not pronounced and likely not reliable for solving this specific problem. Therefore, we conduct experiments based on the state-of-the-art tracker CenterTrack [4], which tracks objects as points and focuses on associating objects locally between consecutive frames with little emphasis on visual features.

As a basis for our experiments, we use two pre-trained models provided by the authors of CenterTrack and fine-tune on top of them to reduce training time and minimize the potential of overfitting. We name the base models CT-COCO and CT-ImNet, where CT-COCO has been pre-trained on the MS COCO dataset [46] and CT-ImNet has been pre-trained on the ImageNet dataset [47]. Both models have a similar architecture with a DLA-34 [48] backbone. We train all our models with a batch size of 12 and a learning rate of 1.25e-4 and we resize and pad the BrackishMOT images from 1920x1080 to 960x544 following the strategy proposed by the CenterTrack authors.

First, we evaluate how pre-training on the two large-scale datasets MS COCO and ImageNet affects CenterTrack's ability to learn from the BrackishMOT sequences. We then extend these results by introducing training strategies for including synthetic sequences, to investigate the potential benefits of using synthetic tracking data in underwater environments. We evaluate all our models based on conventional MOT performance metrics like the Multiple Object Tracking Accuracy metric (MOTA) from the CLEAR MOT metrics [49], the ID F1 score (IDF1) [50], and the recent Higher Order Tracking Accuracy metric (HOTA) [51].

Table E.1: Performance of the CenterTrack models fine-tuned on the BrackishMOT train sequences and evaluated on the BrackishMOT test split.

Model	HOTA↑	MOTA↑	IDF1↑	Dets	GT dets	IDs	GT IDs	IDSW
CT-COCO-Brack	0.36	0.37	0.39	10270	14670	887	182	493
CT-ImNet-Brack	**0.38**	**0.43**	**0.44**	10056	14670	755	182	464

Training on Real Data

We fine-tune the base models on the BrackishMOT train split, which consists of 78 sequences, for 30 epochs and name the new models CT-COCO-Brack and CT-ImNet-Brack. Evaluating these models on the 20 sequences of the BrackishMOT test split gives us an indication of the performance to be expected from fine-tuning a state-of-the-art tracker on manually annotated real data. The HOTA, MOTA, and IDF1 results are presented in Table E.1 along with detections (*Dets*), ground truth detections (*GT dets*), IDs, ground truth IDs (*GT IDs*), and ID switches (*IDSW*).

We see that both models deliver promising results although the model pre-trained on ImageNet outperforms the model pre-trained on MS COCO. This indicates that ImageNet is better suited as a foundation for detecting and tracking objects in this type of underwater environment. We will investigate whether this is also the case when including synthetic data in the following sections.

Training on Synthetic Data

Next, we investigate whether the base models can be taught to track fish in real sequences if they are fine-tuned strictly on synthetic data. We do this by studying how the combinations of the environment with turbidity (T), background (B), and distractors (D) affect the tracking performance. We fine-tune the base models for 10 epochs on the eight different sets of synthetic sequences. The synthetic sequences only contain the *small fish* class, therefore, we evaluate the fine-tuned models on a sub-set of the BrackishMOT test split consisting of the sequences with the *small fish* class. We name this sub-set the 'small fish split' and it contains eight sequences (the list of sequences is presented in Table E.4 as part of another evaluation).

The results of the synthetically trained models evaluated on the small fish split are presented in Table E.2 along with the results of the CT-COCO-Brack and CT-ImNet-Brack models for comparison. Although the synthetic models only perform up to half as well as the models trained on real data, we see that it is in fact possible to train CenterTrack to be able to detect and track the *small fish* class without ever seeing real images of the class. The feature that seems to increase the tracking performance the most is by adding background videos

Table E.2: Performance of CenterTrack models trained strictly on variations of the synthetic dataset. The models have been evaluated on the small fish split.

CT-COCO	HOTA↑	MOTA↑	IDF1↑	**CT-ImNet**	HOTA↑	MOTA↑	IDF1↑
Synth	0.08	-0.17	0.07	Synth	0.08	-0.90	0.06
Synth$_B$	0.12	-0.21	0.12	Synth$_B$	0.14	0.05	0.17
Synth$_T$	0.08	-2.02	0.06	Synth$_T$	0.06	-0.93	0.04
Synth$_D$	0.09	-0.12	0.09	Synth$_D$	0.08	0.03	0.08
Synth$_{TD}$	0.12	-0.14	0.12	Synth$_{TD}$	0.06	0.02	0.05
Synth$_{BT}$	0.19	0.16	0.21	Synth$_{BT}$	**0.19**	-0.24	**0.21**
Synth$_{BD}$	0.15	0.08	0.16	Synth$_{BD}$	0.17	**0.13**	0.19
Synth$_{BTD}$	**0.21**	**0.18**	**0.24**	Synth$_{BTD}$	0.13	0.00	0.14
CT-COCO-Brack	0.37	0.47	0.43	CT-ImNet-Brack	0.39	0.50	0.46

whereas the turbidity and distractors give mixed results. We see that the CT-COCO model performs the best when fine-tuned on the Synth$_{BTD}$ sequences while it is more unclear what benefits the CT-ImNet model the most, however, a good compromise seems to be the Synth$_{BD}$ sequences.

Two Strategies for Training on both Synthetic and Real Data

Previously, we found the synthetic sequences best suited for teaching the base models to track the *small fish* class. Now, we examine whether the CT-COCO model fine-tuned on Synth$_{BTD}$ and the CT-ImNet model fine-tuned on Synth$_{BD}$ provide better foundations compared to the base models. We fine-tune on top of these models for 30 epochs on the BrackishMOT train sequences and name these two-step fine-tuned models CT-COCO-Synth$_{BTD}$ and CT-ImNet-Synth$_{BD}$. Additionally, we examine the potential benefits of combining real and synthetic data in a single training step by fine-tuning the base models for 30 epochs on a combination of the BrackishMOT train and Synth$_{BTD}$ sequences. We name these the CT-COCO-Mix and CT-ImNet-Mix models.

Baseline results for the models are presented in Table E.3 for both the regular test split and the small fish split. We use the small fish split to examine whether the models trained on the synthetic data overfits to the *small fish* class. Generally, we see a tendency that the ImageNet pre-trained models perform

Table E.3: Baseline tracking results for the BrackishMOT test split and the small fish split.

Model		HOTA↑	MOTA↑	IDF1↑		HOTA↑	MOTA↑	IDF1↑
CT-COCO-Brack	**Test split**	0.36	0.37	0.39	**Small fish split**	0.37	0.47	0.43
CT-COCO-Synth$_{BTD}$		0.36	0.38	0.39		0.39	0.47	0.44
CT-COCO-Mix		0.36	0.37	0.39		0.37	0.46	0.43
CT-ImNet-Brack		0.38	0.43	0.44		0.39	0.50	0.46
CT-ImNet-Synth$_{BT}$		0.38	0.42	0.41		**0.41**	**0.52**	0.48
CT-ImNet-Mix		**0.40**	**0.44**	**0.45**		**0.41**	**0.52**	**0.49**

better than the models pre-trained on the MS COCO dataset, which indicates that the ImageNet dataset lays a stronger foundation for detecting the objects of the BrackishMOT dataset. Furthermore, the CT-COCO models do not seem to benefit from the synthetic data, which is in contrast to the results presented in Table E.2 that showed that fine-tuning the CT-COCO model on the Synth_{BTD} sequences gave the best performing purely synthetically trained tracker.

For the CT-ImNet-Synth_{BT} model we see a slight decrease in MOTA and IDF1 when evaluating on the test split, but an increase in the *small fish* sequences, this indicates that the model learns from the synthetic data to better track the *small fish* objects but at the expense of some of the other classes. The CT-ImNet-Mix model exhibits similar performance as the CT-ImNet-Synth_{BT} model on the *small fish* sequences. However, the performance is also increased when looking at all the test sequences, which indicates that the ability to track the other classes is maintained using this training strategy.

4.1 Qualitative Evaluation

In the previous evaluation, we found the overall best-performing model to be CT-ImNet-Mix. In this section, we analyse how the model performs on each of the eight sequences from the small fish split. The qualitative results of the CT-ImNet-Mix model when evaluated on the small fish split are presented in Table E.4. When we inspect the brackishMOT-93 and brackishMOT-95 sequences we see that they have 45 and 1 GT IDs, respectively. However, both sequences score a HOTA performance of 0.44 indicating that a higher number of objects does not seem to have a significantly negative impact on the tracking performance. If we look at BrackishMOT-67 it has four GT IDs but the tracker only manages to get a HOTA score of 0.18. Inspecting the BrackishMOT-67 sequence visually shows that it contains a single medium-sized object of the *small fish* class, which the model tracks well throughout the sequence, however, there are also three tiny objects of the *small fish* class near the seabed that the model largely fails to detect and this penalizes the tracking performance greatly.

Table E.4: Performance of CT-ImNet-Mix model on the sequences of the small fish split.

Sequence	HOTA↑	MOTA↑	IDF1↑	Dets	GT dets	IDs	GT IDs	IDSW
brackishMOT-50	0.40	0.46	0.47	1785	2129	105	17	50
brackishMOT-55	0.35	0.46	0.43	2318	3192	187	37	112
brackishMOT-56	0.53	0.41	0.75	80	87	7	1	4
brackishMOT-67	0.18	0.11	0.10	173	636	31	4	7
brackishMOT-90	0.61	0.53	0.74	401	426	18	3	7
brackishMOT-93	0.44	0.51	0.51	1450	1567	160	45	82
brackishMOT-95	0.44	0.38	0.39	619	148	28	1	0
brackishMOT-98	0.49	0.58	0.60	1728	1930	137	36	80

5 Conclusion

We propose a new underwater multi-object tracking dataset named BrackishMOT, which is an extension of the Brackish dataset captured in turbid waters in Denmark. This is the first and only dataset of its kind and it is a necessary step towards increasing the capability of underwater trackers as there currently only exist very few underwater tracking datasets and they have all been captured in clear tropical waters. Furthermore, we propose a framework for generating synthetic underwater MOT sequences and present baseline results based on fine-tuning CenterTrack using three different training strategies. We show that tracking performance can be increased by including sequences generated by the proposed synthetic framework in the training procedure.

References

[1] United Nations, "Life below water," https://www.un.org/sustainabledevelopment/goal-14-life-below-water/.

[2] N. Madsen, M. Pedersen, K. T. Jensen, P. R. Møller, R. E. Andersen, and T. B. Moeslund, "Fishing with c-tucs (cheap tiny underwater cameras) in a sea of possibilities," vol. 16, no. 2, pp. 19–30, 2021. [Online]. Available: https://www.thejot.net/article-preview/?show_article_preview=1250

[3] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2019, pp. 941–951.

[4] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Computer Vision – ECCV 2020*. Springer, 2020, pp. 474–490.

[5] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *International Journal of Computer Vision (IJCV)*, vol. 129, no. 11, pp. 3069–3087, Sep. 2021.

[6] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "ByteTrack: Multi-object tracking by associating every detection box," in *Computer Vision – ECCV 2022*. Springer, 2022, pp. 1–21.

[7] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2012, pp. 3354–3361.

[8] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking," *arXiv:1504.01942*, apr 2015.

[9] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv:1603.00831*, mar 2016.

[10] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "Mot20: A benchmark for multi object tracking in crowded scenes," 2020.

[11] H. Yu, G. Li, W. Zhang, Q. Huang, D. Du, Q. Tian, and N. Sebe, "The unmanned aerial vehicle benchmark: Object detection, tracking and baseline," *International Journal of Computer Vision (IJCV)*, vol. 128, no. 5, pp. 1141–1159, Dec. 2019.

[12] F. Romero-Ferrero, M. G. Bergomi, R. C. Hinz, F. J. H. Heras, and G. G. d. Polavieja, "idtracker.ai: tracking all individuals in small or large collectives of unmarked animals," *Nature Methods*, vol. 16, no. 2, pp. 179–182, jan 2019.

[13] M. Pedersen, J. B. Haurum, S. Hein Bengtson, and T. B. Moeslund, "3d-zef: A 3d zebrafish tracking benchmark dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2423–2433.

[14] R. B. Fisher, Y.-H. Chen-Burger, D. Giordano, L. Hardman, and F.-P. Lin, *Fish4Knowledge: Collecting and Analyzing Massive Coral Reef Fish Video Data*. Springer International Publishing, 2016, vol. 104.

[15] D. Giordano, S. Palazzo, and C. Spampinato, "Fish tracking," in *Fish4Knowledge: Collecting and Analyzing Massive Coral Reef Fish Video Data*. Springer International Publishing, 2016, pp. 123–139.

[16] L. Kezebou, V. Oludare, K. Panetta, and S. S. Agaian, "Underwater object tracking benchmark and dataset," in *Proceedings of the IEEE International Symposium on Technologies for Homeland Security (HST)*. IEEE, Nov. 2019.

[17] K. Panetta, L. Kezebou, V. Oludare, and S. Agaian, "Comprehensive underwater object tracking benchmark dataset and underwater image enhancement with GAN," *IEEE Journal of Oceanic Engineering*, vol. 47, no. 1, pp. 59–75, Jan. 2022.

[18] T. Mandel, M. Jimenez, E. Risley, T. Nammoto, R. Williams, M. Panoff, M. Ballesteros, and B. Suarez, "Detection confidence driven multi-object tracking to recover reliable tracks from unreliable detections," *Pattern Recognition*, vol. 135, p. 109107, 2023.

[19] M. de Oliveira Barreiros, D. de Oliveira Dantas, L. C. de Oliveira Silva, S. Ribeiro, and A. K. Barros, "Zebrafish tracking using YOLOv2 and kalman filter," *Scientific Reports*, vol. 11, no. 1, Feb. 2021.

[20] H. Wang, S. Zhang, S. Zhao, Q. Wang, D. Li, and R. Zhao, "Real-time detection and tracking of fish abnormal behavior based on improved YOLOV5 and SiamRPN++," *Computers and Electronics in Agriculture*, vol. 192, p. 106512, Jan. 2022.

[21] M. Pedersen, N. Madsen, and T. B. Moeslund, "No machine learning without data: Critical factors to consider when collecting video data in marine environments," vol. 16, no. 3, 2021. [Online]. Available: https://www.thejot.net/archive-issues/?id=73

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 25. Curran Associates, Inc., 2012, pp. 1097–1105.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

[24] J. Jäger, V. Wolff, K. Fricke-Neuderth, O. Mothes, and J. Denzler, "Visual fish tracking: Combining a two-stage graph approach with CNN-features," in *OCEANS 2017 - Aberdeen*. IEEE, Jun. 2017.

[25] T. Liu, P. Li, H. Liu, X. Deng, H. Liu, and F. Zhai, "Multi-class fish stock statistics technology based on object classification and tracking algorithm," *Ecological Informatics*, vol. 63, p. 101240, Jul. 2021.

[26] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020.

[27] M. A. M. Martija and P. C. Naval, "SynDHN: Multi-object fish tracker trained on synthetic underwater videos," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*. IEEE, Jan. 2021.

[28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[29] Y. Xu, A. Ošep, Y. Ban, R. Horaud, L. Leal-Taixé, and X. Alameda-Pineda, "How to train your deep multi-object tracker," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[30] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[31] A. Mahmood, M. Bennamoun, S. An, F. Sohel, F. Boussaid, R. Hovey, and G. Kendrick, "Automatic detection of western rock lobster using synthetic data," *ICES Journal of Marine Science*, vol. 77, no. 4, pp. 1308–1317, Nov. 2019.

[32] M. Bewley, A. Friedman, R. Ferrari, N. Hill, R. Hovey, N. Barrett, O. Pizarro, W. Figueira, L. Meyer, R. Babcock, L. Bellchambers, M. Byrne, and S. B. Williams, "Australian sea-floor survey data, with images and expert annotations," *Scientific Data*, vol. 2, no. 1, p. 150057, oct 2015.

[33] J. Musić, S. Kružić, I. Stančić, and F. Alexandrou, "Detecting underwater sea litter using deep neural networks: An initial study," in *Proceedings of the International Conference on Smart and Sustainable Technologies (SpliTech)*. IEEE, Sep. 2020.

[34] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH '87*. ACM Press, 1987.

[35] K. Stephens, B. Pham, and A. Wardhani, "Modelling fish behaviour," in *Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*. ACM, Feb. 2003.

[36] C. Hartman and B. Beneš, "Autonomous boids," *Computer Animation and Virtual Worlds*, vol. 17, no. 3-4, pp. 199–206, 2006.

[37] S. Podila and Y. Zhu, "Animating escape maneuvers for a school of fish," in *Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. ACM, Feb. 2017.

[38] Y. Ishiwaka, X. S. Zeng, M. L. Eastman, S. Kakazu, S. Gross, R. Mizutani, and M. Nakada, "Foids," *ACM Transactions on Graphics*, vol. 40, no. 6, pp. 1–15, Dec. 2021.

[39] M. Pedersen, J. Bruslund Haurum, R. Gade, and T. B. Moeslund, "Detection of marine animals in a new underwater dataset with varying visibility," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 18–26.

[40] M. Pedersen, J. B. Haurum, P. Dendorfer, and T. B. Moeslund, "MOTCOM: The multi-object tracking dataset complexity metric," in *Computer Vision – ECCV 2022*. Springer, 2022, pp. 20–37.

[41] U. Technologies, "Unity," 2005, accessed: 2023-02-21. [Online]. Available: https://www.unity.com

[42] Y. Kano, M. S. Adnan, C. Grudpan, J. Grudpan, W. Magtoon, P. Musikasinthorn, Y. Natori, S. Ottomanski, B. Praxaysonbath, K. Phongsa, A. Rangsiruji, K. Shibukawa, Y. Shimatani, N. So, A. Suvarnaraksha, P. Thach, P. N. Thanh, D. D.

Tran, K. Utsugi, and T. Yamashita, "An online database on freshwater fish diversity and distribution in mainland southeast asia," *Ichthyological Research*, vol. 60, no. 3, pp. 293–295, Jun. 2013.

[43] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org

[44] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017.

[45] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2018.

[46] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Computer Vision – ECCV 2014*. Cham: Springer, 2014, pp. 740–755.

[47] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2009, pp. 248–255.

[48] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018.

[49] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, pp. 1–10, May 2008.

[50] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Computer Vision – ECCV 2016 Workshops*. Springer, 2016, pp. 17–35.

[51] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *International Journal of Computer Vision (IJCV)*, vol. 129, no. 2, p. 548–578, oct 2021.

Paper F

No Machine Learning Without Data: Critical Factors to Consider when Collecting Video Data in Marine Environments

Malte Pedersen, Niels Madsen, and Thomas B. Moeslund

The paper has been published in the
The Journal of Ocean Technology, 2021, as an *essay*.

The layout has been revised.

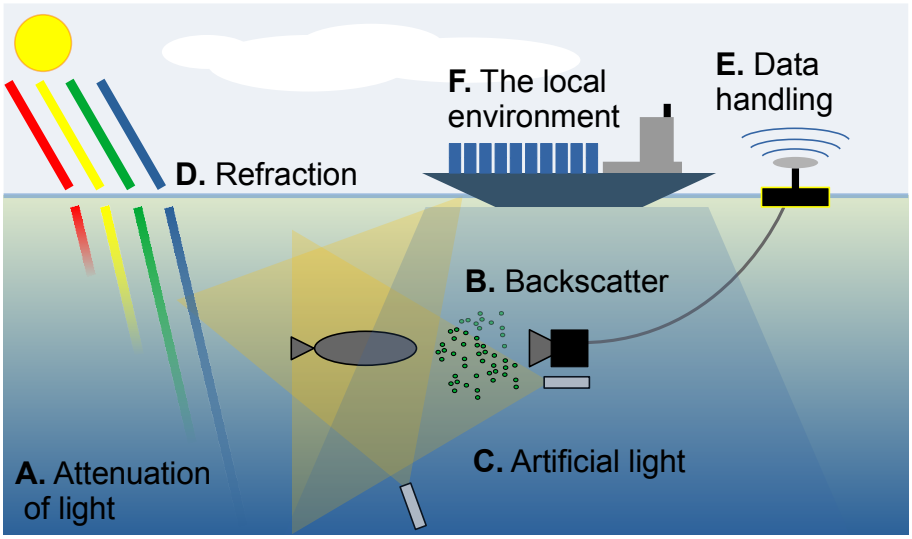


Fig. F.1: Illustration of six important factors to take into account when collecting video data from underwater environments.

1 Introduction

An increased political focus on the condition of our marine ecosystems has put researchers under pressure to gather and analyze data at an unprecedented pace. Assessing the impact of global climate change, pollution, and overfishing on the biodiversity and fish stocks are major challenges for fisheries and governments across the world. An increasingly popular tool for gathering biological data in a non-intrusive manner is automated analysis of image and video data using computer vision and machine learning. However, large-scale image based data collection and automated analysis has not traditionally been common practice among marine researchers.

While images of a given object captured in air looks more or less the same independent of where on the planet you take the photo, this is not the case in marine environments. Images are formed in a camera by capturing the light reflected from objects within the camera's field of view, but marine waters are filled with organic and inorganic matter that absorbs and scatters light. This causes the visibility and coloring to vary widely depending on the location, time, depth, and weather which can make it a challenge to capture high quality video recordings of underwater objects; and without sufficient high quality data, any machine learning algorithm will fail.

Machine learning and computer vision is increasingly used within several fields of biology, but there seem to be a hesitancy when it comes to under-

water research areas such as fisheries and marine science. The main reason being that, traditionally, it has been extremely demanding and expensive to capture high quality underwater video footage suitable for automated analysis. However, during the past decade the price on conventional action and underwater cameras has dropped substantially while the image quality has increased. Moreover, the performance of state-of-the-art computer vision and machine learning algorithms has sky-rocketed during the same period, with the introduction of deep neural networks.

Neural networks are machine learning algorithms that learn in a way somewhat similar to the way children learn. They need to see things many times in different settings and be told what they are looking at to be able to distinguish between them. By presenting a deep neural network to large numbers of images of fish, it is possible for the network to learn how to detect and distinguish between species, e.g., mackerels, sardines, cod, and tuna. Another network may be designed to track fish through video sequences which can be used for behavioral analysis or for controlling a by-catch release-mechanism inside a trawl. There are many possibilities of using deep neural networks for automating processes in marine settings, but independent on the task at hand or the choice of network, there is a demand for annotated data.

In this essay we present and discuss the main factors that influence the data capturing process. We hope this will pave the way for other marine researchers to capture high quality data and thereby set the stage for using machine learning algorithms in marine monitoring.

2 The Six Factors

It is not feasible to create a single protocol for underwater video monitoring due to the extreme variations in marine environments across the globe. However, the six factors illustrated in Figure F.1 (attenuation of light, backscatter, artificial light, refraction, data handling, and the local environment) should always be taken into account when capturing data for a machine learning based marine video monitoring system. In the following, each of the factors will be discussed in greater details.

2.1 Attenuation of Light

Probably the most significant difference between capturing images in air and water is caused by the attenuation of light. While it is most often not necessary to take attenuation of light in air into account, it is another matter in water. As light enters water, it takes only a few meters before the long wavelength colors in the red spectrum are absorbed by the water. This is followed by the absorption of the yellow, green, and lastly blue wavelength colors.

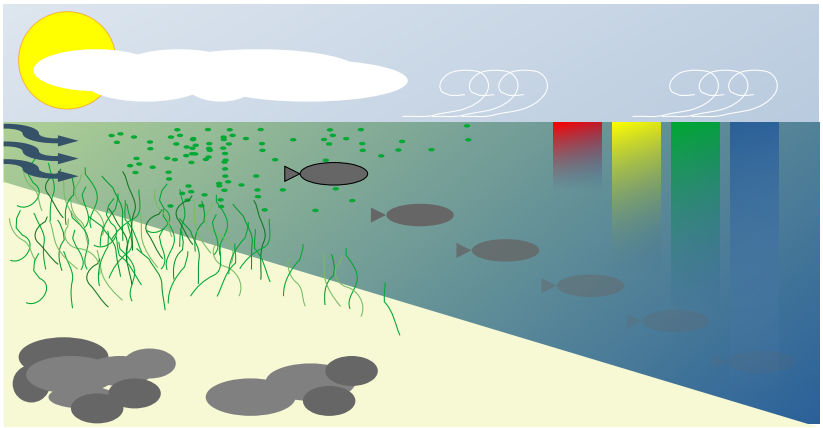


Fig. F.2: Coastal areas can be especially nutrient rich with a high concentrations of phytoplankton that makes the water appear green or brownish. Furthermore, sediments can be resuspended in shallow waters due to the currents and wind. As the depth increases the water gets more clear, but the light intensity is reduced.

The exact depths at which the wavelengths of natural light are absorbed in our oceans, estuaries, and rivers vary greatly dependent on the intensity of the light, particles in the water, and other factors. However, objects that are observed through natural light will always appear more dim and colorless as the depth increases, as illustrated in Figure F.2.

It is particularly difficult to capture high quality recordings in coastal waters and estuaries as the visibility can vary to a large degree. An example of varying degrees of visibility in a shallow strait is presented in Figure F.3. Low visibility is often due to resuspension of sediments and eutrophication caused by shallow water, wind, and excessive amounts of nutrients. Ecosystems with high concentrations of phytoplankton appear green or brownish due to the chlorophyll and carotenoid pigments that reflect green and orange-red wavelengths, respectively.

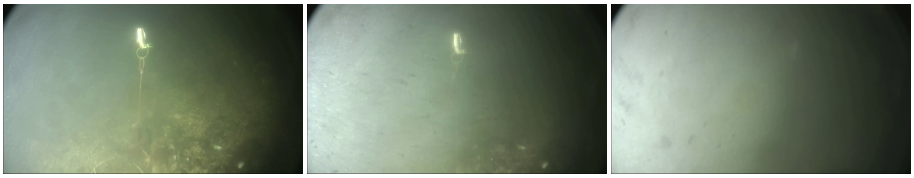


Fig. F.3: Three photos captured from a stationary camera a few minutes apart in a brackish strait at 9m depth with artificial light. The images appear brownish due to high numbers of phytoplankton and resuspended sediments. The visibility goes from semi-clear to unclear from left to right. In shallow straits, estuaries, and coastal areas the water can turn unclear rapidly and is rarely clear at any point.

2.2 Backscatter

Artificial light can be used to counteract low visibility caused by the attenuation of light. However, there are several things to be aware of when using artificial light in water. The placement and direction of the light can introduce backscatter, which is light absorbed and scattered by small particles in the water between the lens and the object. Backscatter can be the cause of significant noise and it can occur even in seemingly clear water.

The water is semi-clear in Figure F.4, but it is difficult to see the sea bed due to strong backscatter caused by a single artificial light placed close beneath the camera and pointing into the water column in front of the camera's field of view. The backscatter appears almost like a thick fog with sprinkles due to the varying size of scattering particles in the water. A less severe case can be seen in Figure F.5 where the water is also semi-clear and backscatter occurs as sprinkles in the left side of the image. The single light source is placed to the left of the camera and is illuminating the scene in an indirect manner allowing for a more clear view of the sea bed.

In environments where the water is mostly unclear it may not be suitable to use a conventional camera due to the short visual range. Specialized sensors, such as range-gated time-of-flight (ToF) cameras or sonars, can be used to minimize the effect of backscatter and obtain information of objects not seemingly visible. A ToF camera can measure the distance between the camera and the objects in a scene using active illumination and measuring the time it takes from the light is emitted and until it is received by the image sensor. Range-gating allows the ToF camera to only open the shutter and receive light that has traveled a given distance, which is an effective way to see past backscatter. The visual distance of range-gated ToF cameras is, however, still dependent on that a certain amount of artificial light penetrates the turbid water and reaches the object. Moreover, the resolution of ToF cameras is most often lower than for conventional cameras. In scenes with very unclear water even range-gated cameras fall short and an alternative can be to use sonar. Sonars are sound-

Fig. F.4: Strong backscatter caused by artificial light positioned close to the camera in semi-clear water.

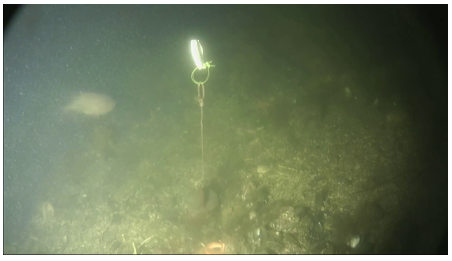


Fig. F.5: Sprinkled backscatter in the left side of the view in semi-clear water caused by indirect artificial light.

based sensors capable of gathering depth information across long distances independent of the water clarity. However, sonars generally have a very low resolution which makes it nearly impossible to recognize species, count the number of objects, and similar.

2.3 Artificial Light

Recording high resolution underwater color videos is currently only possible using conventional cameras, but as depth increases so does the attenuation of light. The reduction in light causes objects to appear dim and featureless and artificial light can be a necessity. However, it is a non-trivial task to place the light source in an optimal position and it is highly dependent on the environment.

In clear and non-scattering water it may be possible to illuminate an entire scene satisfactory using a single source of light placed close to the camera, see Figure F.6a. However, this setup can be the cause of strong backscatter even in slightly turbid water, see Figure F.6b. Backscatter can be minimized by placing the light further away from the camera and in an angle, but this may cause uneven illumination and shadows, see Figure F.6c. Multi-directional illumination is a way to combat the problems of uneven lighting and shadows but it requires a larger and more complex setup, see Figure F.6d.

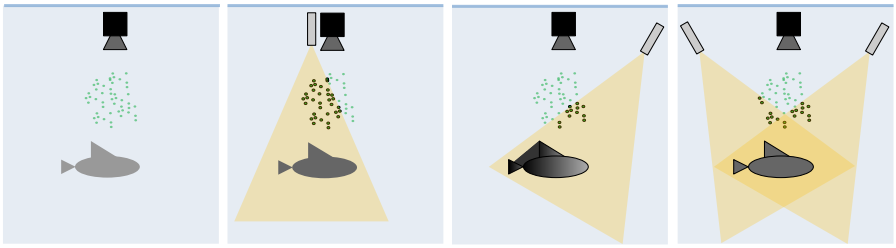


Fig. F.6: Four light setups for capturing underwater images. (a) No artificial light; the object appears dim and colorless. (b) A single light; even illumination, but significant backscatter. (c) The light is placed at an angle; backscatter is minimized, but shadows and uneven illumination occur. (d) Two light sources; backscatter is minimized and even illumination is achieved.

It is also possible to reduce backscatter even further by placing the light source very close to the object, but here it is extremely important to take into consideration whether non-uniform lighting, over exposure, and shadows can be a problem as seen in Figure F.7. Generally, the exercise is to find the best trade-off between an even illumination and a minimum amount of backscatter.

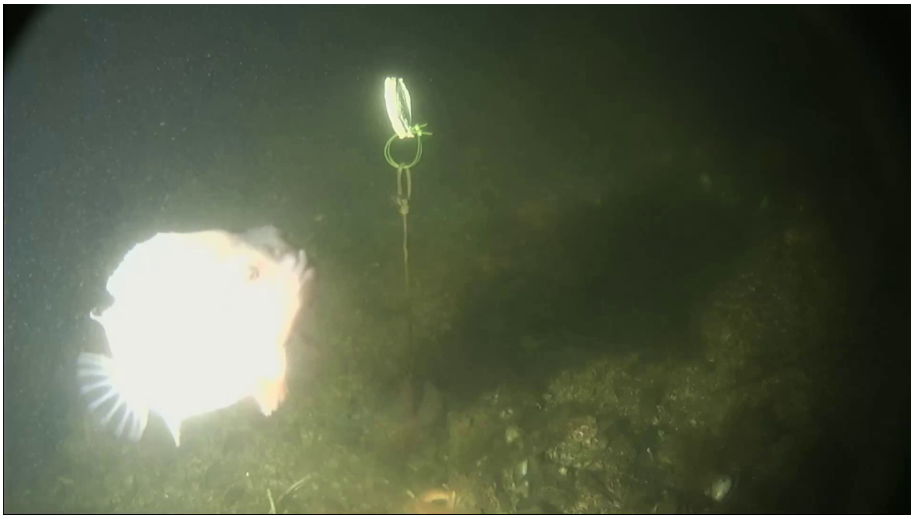


Fig. F.7: A single light source is placed to the left of the camera at an angle. Some sprinkled backscatter is seen in the left part of the image. The fish is swimming close to the light source causing overexposure and strong shadows on the sea bed.

2.4 Refraction

The speed of light varies depending on the medium it travels through. When light moves from air to water the speed is slowed down and this causes an effect known as refraction, where the direction of the light changes with respect to the incident angle and velocity. Refraction of light can be described by Snell's Law given by

$$\frac{sin\theta_f}{sin\theta_t} = \frac{v_f}{v_t} = \frac{n_t}{n_f}, \tag{F.1}$$

where θ is the angle between the surface normal and the light ray, v is the velocity, n is the refraction index, and the subscripted f and t indicate the two media the light travels from and to, respectively. An example of how refraction affects an image can be seen in Figure F.8. The black lines illustrate the rays with no refraction, whereas the dotted lines show the directional change caused by refraction.

Depending on the type of camera, lens, and underwater-casing refraction can significantly decrease the accuracy of measurements as it alters the intrinsic parameters of the camera and distorts the image. A commonly used method to minimize the effect of refraction is to take basis in the perspective camera model and calibrate the camera using a checkerboard or calibration frame. This can provide relatively accurate results if the scene of interest is restricted to a limited space, however, it has been shown that refraction causes single view point models to be invalid and the error grows concurrent with distance

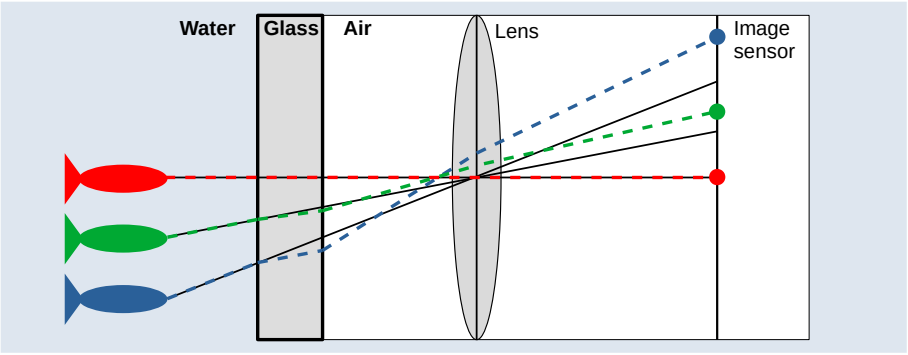


Fig. F.8: The dotted lines illustrate the refracted light rays, whereas the black lines illustrate the paths the rays would travel through air alone. Notice how the positional displacement on the image sensor increases with the incident angle of the ray.

and angle to the image sensor. A more robust method is to model the physical attributes of refraction and use ray tracing to adjust for the errors introduced by the perspective model.

Choosing the right calibration method can be difficult and it is dependent on the application. Calibration and refraction handling is most often not a necessity for handling relatively simple image processing problems like object detection. However, more complicated machine learning tasks like classification or re-identification may benefit from it. If precise object tracking or 3D reconstruction is the goal, then calibration and refraction handling can be critical.

2.5 Data Handling

Recording videos under water can be extremely demanding and therefore the aim is to get as much out of the recordings as possible. Data storage can be a problem and whether it is long or short term monitoring a goal is often to keep the storage at a minimum. Therefore, it is important to know what type of image analysis is to be conducted on the data. If the task is to count the number of fish using **object detection**, the image resolution and frame rate can be kept relatively low and the videos may even be compressed without significantly influencing the detection rate. Expanding the task to include **classification** requires a higher resolution and for reliable object **tracking** a high frame rate is important as marine organisms can move both fast and erratic. Temporal compression should generally be avoided as it introduces motion blur and amplifies noise, e.g., caused by particles flowing in the water as illustrated in Figure F.9.

For long term monitoring projects it is important to ensure a steady power

Fig. F.9: Video compression can reduce the storage size significantly but it also removes information and introduces noise. In this example, small particles draws semi-transparent lines across the image, due to temporal compression, while flowing from left to right.

supply and a way to retrieve data regularly. Regular data retrieval puts less demands on the storage capability of the internal hardware, allows for routinely inspection of the data, and serves as a vital backup. Floating stations are common for off-shore operations, while cabled land-based stations can be used in some coastal areas.

2.6 The Local Environment

The local environment can play a pivotal role in unforeseeable ways. Here we mention common problems that can hinder an otherwise well-structured underwater monitoring setup.

Algae can bloom on the lens within a few days depending on the environment. Algae growth will cloud the view and make the quality of the recordings poor or even useless. In Figure F.10 the fish and the surroundings appear green due to algae on the lens and phytoplankton in the water.

Permissions from the local municipality or national maritime authorities may be needed to conduct research in wildlife sanctuaries or close to ship traffic, such as harbors or channels.

Ship traffic can easily destroy a floating surface station, therefore, it is crucial to mark observation spots properly. In cold regions it is also vital to consider potential floating ice.

Flickering from the sun light when it hits the waves is especially apparent in shallow waters. Additionally, clouds and boats can cause shadows that will darken the entire scene.

The behavior of marine organisms will in most cases be affected by the presence of cameras, humans, or vehicles unless well hidden. An example can be seen in Figure F.11 where a school of sticklebacks is lingering in front of the camera attracted by a light source. The use of artificial light can both attract

and repel animals and alter the local environment around the setup of long term monitoring projects. Therefore, it is extremely important to take into consideration whether artificial light is necessary for a given setup.

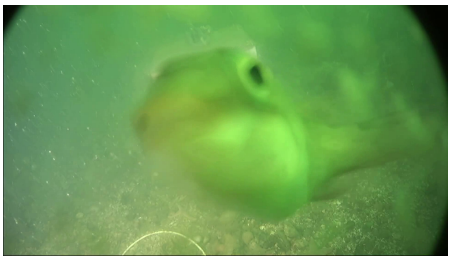


Fig. F.10: Algae on the lens can make everything appear green and will in the worst case block the view entirely. Algae growth varies greatly depending on the local environment.

Fig. F.11: The behavior of marine organisms is strongly affected by the presence of artificial light, such as this school of sticklebacks lingering in front of a light source.

3 Final Remarks

An obstacle of most state-of-the-art machine learning based computer vision solutions is the need for large annotated image datasets. There are very few available underwater datasets compared to the terrestrial counterparts, and this is a hindrance for the development of dedicated marine vision algorithms. The variance between underwater environments can be extreme and it is therefore crucial to have training data from as many regions, environments, and ecosystems as possible to build a strong foundation for the coming generation of marine vision algorithms.

We urge marine researchers to be open-minded about using cameras, marine vision, and machine learning in their research, and sharing their datasets and annotations with the public. Hopefully, the presentation of the six factors can serve as a stepping stone for many future marine image and video data collection tasks to the benefit of the marine research community.

Paper G

Re-Identification of Giant Sunfish using Keypoint Matching

Malte Pedersen, Joakim Bruslund Haurum,
Thomas B. Moeslund, and Marianne Nyegaard

The paper has been published in the
Proceedings of the Northern Lights Deep Learning Workshop, 2022.

The layout has been revised.

Abstract

We present the first work where re-identification of the Giant Sunfish (Mola alexandrini) is automated using computer vision and deep learning. We propose a pipeline that scores an mAP of 60.34% on a full rank of the novel TinyMola dataset which includes 41 IDs and 91 images. The method requires no domain-adaptation or training which makes it especially suited for low-budget or volunteer-based projects, like Match My Mola, as part of a human-in-the-loop model.

The pipeline includes segmentation, keypoint detection and description, keypoint matching, and ranking. The choice of feature descriptor has the largest impact on the performance and we show that the deep learning based SuperPoint descriptor greatly outperforms handcrafted descriptors like SIFT and RootSIFT independent of the segmentation level and matching method. Combining SuperPoint and the graph neural network based SuperGlue matching method produces the best results.

1 Introduction

The world's heaviest bony fish, the elusive 'Giant Sunfish' (*Mola alexandrini*), can reach an impressive weight of 2.3 ton [1]. Globally, they are rarely seen by SCUBA divers, but are frequent seasonal visitors to the Bali area, Indonesia [2]. Here, they seek cleaner fish interaction for removal of skin parasites, and are a highly popular target of the local SCUBA tourism industry [3]. Little is known of this seasonal sunfish phenomenon, including if the tourism is reliant on a small, local sunfish population with high site fidelity, or transient sunfish with low re-visitation rates. Understanding this is critical for assessing the potential impacts (and any need for regulation) of diver crowding, which causes disruptions to sunfish-cleaner fish interactions [2].

To investigate this, the citizen science and volunteer based project, Match My Mola [4], collects and curates sunfish images from the Bali area, taken by tourist divers, for photo identification purposes. Images are manually compared pair-wise to assess re-sightings of individuals over time, however, with increasing image numbers match time becomes a significant challenge to this volunteer-based project, and an automated system is critically needed.

Re-identification has been an active research problem within computer vision for decades. However, like in other fields the research has mainly been focused on humans [5] and only few have taken a glance into the aquatic world [6–9]. In this work we present the first scientific attempt to re-identify sunfish and show that it is possible based on the number of keypoint correspondences as illustrated in Figure G.1.

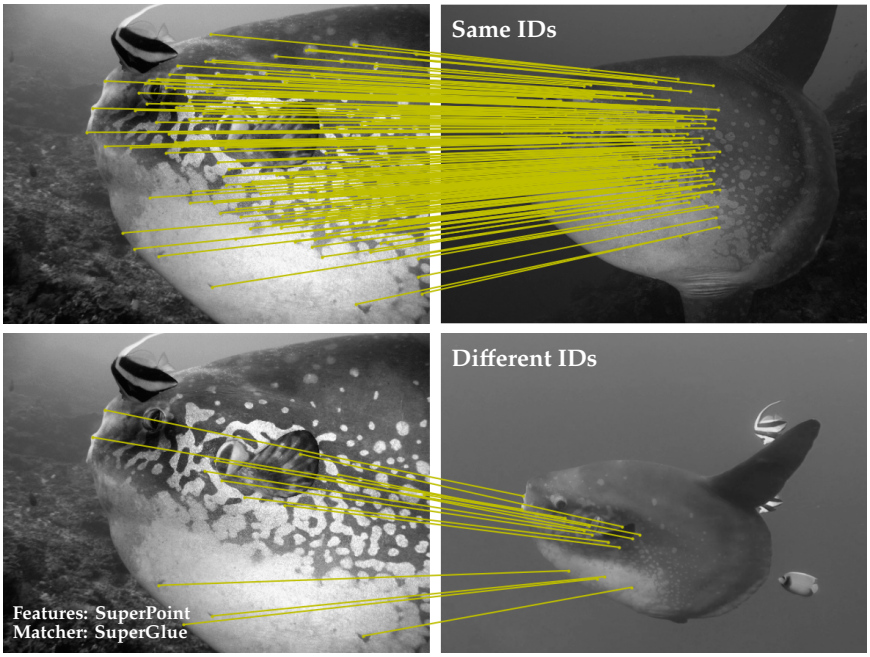


Fig. G.1: Re-identification based on the number of keypoint matches. The images in the top and bottom rows are of the same and different individuals, respectively. The yellow lines connect matched keypoints.

Our contributions include:

1. a re-identification pipeline that requires no domain-adaptation or training.
2. an evaluation of how the segmentation level affects the performance of the system.
3. a comparison between the handcrafted feature descriptors SIFT and RootSIFT and the deep learning based SuperPoint feature descriptor.

2 Related Work

Photographic identification has been used for studying wild marine animals in a non-intrusive manner for decades [10, 11]. It allows researchers to identify the same individual across different years, but requires manual labour to obtain the photographs and match the individuals from the captured footage. Citizen science projects have proven to be an effective and irreplaceable method to gather large amounts of data. But as the databases grow, so does the need

for manual labor. Therefore, computer vision has become an essential tool to scale such research.

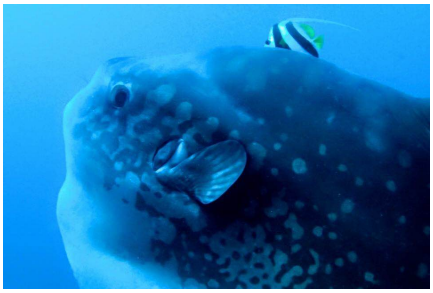
Image processing and pattern matching techniques have been used to automatically identify individuals of whale-sharks [6], spotted raggedtooth sharks [12], and patterned terrestrial animals [7, 13]. However, recently Siamese networks and the use of triplet loss have become a popular means for handling re-identification problems within marine vision. Wang et al. proposed to use a Siamese network and adversarial training to identify whales by their flukes [14] and Nepovinnykh et al. trained a Siamese network for Ringed Seal re-identification [15]. Deep learning has generally gained a lot of attention during the last decade, which is also seen in the work done by Bouma et al. where they train a ResNet50 model using a triplet loss for identifying dolphins by their fin tail [16]. A ResNet50 was also used by Moskvyak et al. in their work on re-identification of manta rays [8, 17], where they proposed to embed the feature vectors by body landmark information and use a weighted combination of three losses. On a higher level, Schneider et al. investigated how the performance of CNNs was affected by using either Siamese networks or triplet loss for animal re-identification and found that triplet loss generally outperforms Siamese networks [18].

A common issue with the aforementioned methods is the requirement for training data and domain adaptation. However, it is demanding to capture images in wild underwater environments and marine image datasets are, therefore, often sparse. This leaves little to no room for the creation of high quality data splits.

3 TinyMola Image Dataset

The dataset used in this work is named 'TinyMola' and it is a subset of the much larger Match My Mola image database, which consists of more than a thousand photo events (PhE). PhEs are 1-3 images of the same individual captured by the same diver during the same dive and the images can be of one or both sides of the fish as illustrated in Figure G.2. Manually identifying sunfish between PhEs is both hard and time-consuming and only 29 individuals have been matched and verified by the marine scientists at this point. These individuals form the basis of the TinyMola dataset as no ground truth is available for the remainder of the dataset.

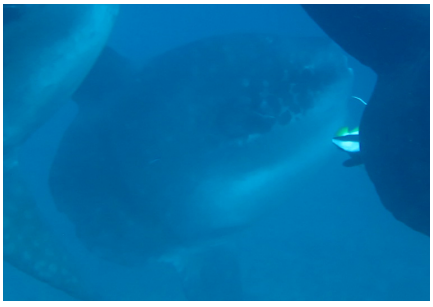
The sunfish have unique markings which are used to identify the individuals. However, the markings on the fish are not identical on the two sides and they cannot be directly compared. Therefore, we frame the re-identification task to be side-specific and provide each side of the fish a unique ID. For each ID there are images from at least two PhEs. However, there are cases where two PhEs of the same individual include images from both sides in one of



(a) Left side of the fish.

(b) Right side of the fish.

Fig. G.2: Two images from the same photo event. The images are captured at different distances and angles, and shows both sides of the fish.



(a) Low contrast.

(b) Extreme angle.

Fig. G.3: Image variations. Sometimes the objects are only partly visible, the contrast may be low, or the object is captured from an extreme angle.

the PhEs but not in the other. These 'unpaired' images are named *singles* if they do not match with images from any other PhE. We have a total of 41 IDs shared among 14 left-side, 17 right-side, and 10 single IDs as summarized in Table G.1.

The quality of the images vary extensively depending on the turbidity of the water, attenuation of light, occlusion, and camera settings [19]. Two examples illustrating some of the variations can be seen in Figure G.3. The resolution of the images varies from 0.1 to 16 megapixels (MP) with a mean around 4 MP and an object resolution around 1 MP on average. To standardize the images we resize them to a resolution of 640x480 and convert them to gray-scale.

Table G.1: TinyMola Dataset composition.

	Left	Right	Singles	Total
IDs	14	17	10	41
Images	37	44	10	91

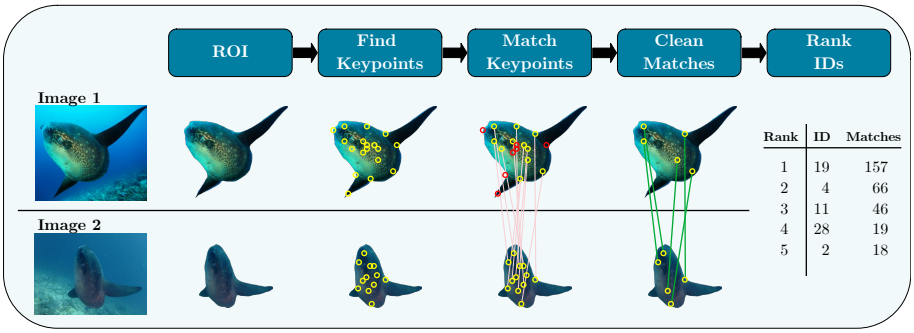


Fig. G.4: The proposed pipeline. In the first module the ROI is either the full image, bounding box, or instance segmentation. Next, keypoints are located (illustrated as yellow circles). During matching some keypoints are not paired (represented by red circles). Ambiguous or weak keypoints are dismissed during cleaning and lastly the images are ranked based on the number of keypoint matches.

4 Method

As previously mentioned, re-identification of sunfish is currently conducted manually by marine researchers. The researchers crop the image around the target and look at markings across all overlapping body parts on the two images. If the markings are barely visible the image may be subject to contrast enhancement. The images are then compared pair-wise to images of other sunfish and matches are noted and examined by other matching-experts, to confirm that the images are of the same individual.

We propose a solution inspired by the manual process for ranking the images based on the number of matching keypoints. The pipeline is illustrated in Figure G.4 and the modules are described below.

4.1 Region of Interest

We want to investigate whether cropping the image has an effect on the performance of the system. Therefore, we evaluate three levels of segmentation: 1) full image, 2) bounding box, and 3) instance segmentation. An ImageNet [20] pre-trained Mask R-CNN 50 FPN model [21] from Detectron2 [22] is used for segmenting the objects. The model is fine-tuned for 300 epochs with a batch size of 6 and a train split consisting of 100 random images of sunfish from the Match My Mola database, which are not part of the TinyMola dataset. The fine-tuned model achieves an average precision of 87.7% on the segmentation task and the bounding boxes are simply drawn around the segmentations.

4.2 Keypoints

The body of a sunfish is highly rigid, except for the dorsal and pelvic fins. Consequently, the markings on the fish are mainly affected by affine transformations such as rotation and scale. We test and evaluate the performance of two handcrafted feature descriptors (SIFT and RootSIFT) and one state of the art deep learning based feature descriptor (SuperPoint), which are all summarized here.

The Scale Invariant Feature Transform (SIFT) keypoint descriptor was proposed by Lowe [23, 24] and has been among the most popular keypoint descriptors for two decades. Interest points are located in the image by creating a scale-space using difference-of-Gaussians and finding consistent extrema points. A histogram of oriented gradients (HoG) with 36 bins is created from a region around the point and used to assign an orientation to the keypoint. The SIFT descriptor itself is based on a 4x4 matrix of normalized HoG features with 8 bins resulting in a feature vector with 128 values.

Originally, Lowe proposed to match SIFT features by Euclidean distance, however, as noted by Arandjelovic and Zisserman [25] the Hellinger kernel has often been used to compare histograms as it commonly yields superior results compared to Euclidean distance. As the SIFT descriptor is based on histograms they proposed an enhanced method named RootSIFT, which consists of two additional steps: 1) L1-normalize the SIFT feature vector and 2) square root each element. Consequently, comparing RootSIFT features using Euclidean distance, is equivalent to compare SIFT features using the Hellinger kernel, which often increases performance.

Recently, deep learning has been used to both detect and describe keypoints. SuperPoint [26] is among the state of the art methods that handles both tasks jointly. SuperPoint is a CNN that has been trained on synthetic data of angular shapes, such as triangles, lines, and cubes. Subsequently, the model is finetuned on images from MS-COCO [27] in a self-supervised manner using homographic adaptation, which is the use of random homographies to learn image-to-image transformations that may appear in real world scenarios. The SuperPoint feature vector has a dimensionality of 256.

We evaluate the performance of all three descriptors with default parameter values and a maximum of 1024 keypoints per image. All the keypoint descriptors are calculated on the full image, but is only part of the matching process if they are located within the ROI.

4.3 Keypoint Matching

Finding corresponding keypoints in two images is a matter of determining which pair of features that are most similar (nearest neighbor) determined by a distance function. In the following we will very briefly describe two

traditional methods (brute-force and kd-trees) and a graph neural network (SuperGlue) for finding the nearest neighbor.

Depending on the problem, and the dimensionality and nature of the data, keypoint matching has commonly been done using brute-force methods or kd-trees [28]. Brute-force methods compare all elements in the two distributions and are guaranteed to find the best match, but the processing time can be high for large distributions. On the other hand, kd-trees do not guarantee to find the best match, but are faster for large distributions. As the TinyMola dataset is small and the task is an offline problem we use brute-force to match the keypoints to get optimal results.

Recently, deep learning has made its entry into keypoint matching and SuperGlue [29], proposed by the team behind SuperPoint, is currently one of the state of the art methods. For each keypoint SuperGlue takes the position and feature descriptor as input and encodes it using a multilayer perceptron. The spatial and visually encoded feature vectors are fed into a graph neural network that utilizes self- and cross-attention to compute matching descriptors. A similarity matrix is computed with added "dustbin" columns and rows to handle non-matched keypoints. Lastly, the Sinkhorn algorithm is used to compute the optimal partial assignment.

The SuperGlue algorithm is designed to be used with SuperPoint which has twice as many elements as SIFT and RootSIFT. Therefore, to make a fair comparison we use brute-force to match the SIFT, RootSIFT, and SuperPoint descriptors. Additionally, we also match SuperPoint features using two pre-trained SuperGlue models: *SGIndoor*, that has been trained on indoor images from ScanNet [30] and *SGOutdoor* that has been trained on a subset of outdoor images from YFCC100M [31].

4.4 Clean Matches

Naively matching the closest keypoints can lead to poor results. For this reason, David G. Lowe introduced the distance ratio test [24] as a way to dismiss keypoints that are ambiguous. If the ratio between the distance to the nearest and second nearest neighbor is above a threshold, the keypoint is considered too uncertain and is discarded. The optimal threshold depends on the nature of the data and if it is too low too many correct matches may be discarded and vice versa.

When using the brute-force method to match the keypoints we clean the matches using the distance ratio test with a threshold of 0.8 as proposed by Lowe [24]. We do not clean the matches proposed by SuperGlue as it dismisses weak and ambiguous candidates through the dustbin and the Sinkhorn assignment scheme.

5 Evaluation

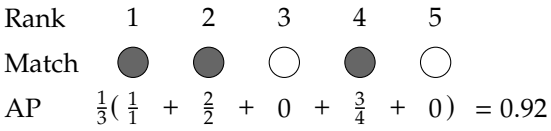
We evaluate the performance of the proposed methods by their mean average precision (mAP) per rank. We view every image of the dataset (except the *single* images) as *probes* and compare each probe against all the other images, which we call the *gallery* images. The *single* images are included in the set of gallery images. There is always at least one gallery image with the same ID as the probe and we name these the *hit* images. We calculate the average precision as

$$\mathrm{AP} = \frac{1}{H}\sum_{n=1}^{k} P_n R_n, \tag{G.1}$$

where H is the number of hit images, k is the rank, P is the precision, and R is a relevance function. The precision is given by

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{G.2}$$

where TP is the number of true positives and FP is the number of false positives. The relevance function, R, takes a value of 1 or 0 depending on whether the match is a hit or not. The rank decides the number of matches to take into account and the matches are sorted in a decreasing manner based on the number of keypoint correspondences between the probe and gallery image. The number of hit images is bounded by the rank such that we have $H \leq k$. An example of calculating the AP is given below where $H = 3$ and $k = 5$. The filled and empty circles represent hits and misses, respectively.



Lastly, the mean AP is given by

$$\mathrm{mAP} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{AP}_i \tag{G.3}$$

where N is the number of probes.

6 Results

The performance of the system is measured by the mAP which is presented against the rank in Figure G.5. There are several interesting aspects that can

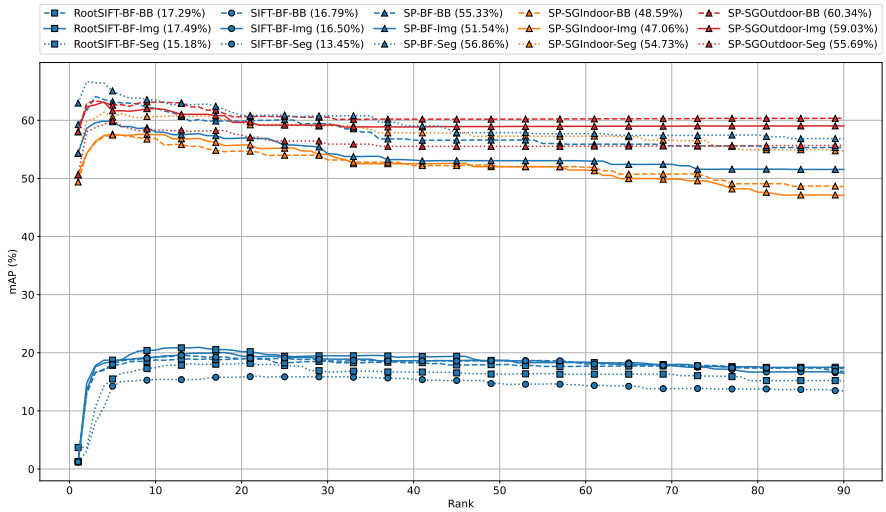


Fig. G.5: Evaluation results. The legends specify *descriptor-matching-segmentation*, e.g., SP-SGIndoor-BB is a combination of the SuperPoint descriptor, SuperGlue indoor matching, and bounding box segmentation. The handcrafted feature descriptors (SIFT and RootSIFT) show weak performance compared to the deep learning based SuperPoint descriptor. The difference between using brute-force matching and the graph-based SuperGlue is less profound and the segmentation level seems to affect the performance ambiguously. The mAP presented in the legends are from the last rank.

be seen from the results. One thing to notice is the significant difference in performance between the feature descriptors. For both of the handcrafted descriptors (SIFT and RootSIFT), the mAP at rank 1 is close to zero indicating that the ranking is more or less based on coincidence. On the other hand, the deep learning based SuperPoint descriptor shows promising results both when using brute-force matching and SuperGlue. Decreasing the region of interest seems to have an ambiguous effect; for RootSIFT, SIFT, and SP-SGOutdoor it generally worsen the performance, but for SP-BF and SP-SGIndoor it increases the performance.

Both SIFT and RootSIFT have few hits on the first rank, but the precision increases up to rank 10 and stabilizes. On the other hand, we see that the SuperPoint based methods all perform well already on rank 1 and their performance increases slightly before dropping and stabilizing, which indicates at least a single hit among the first few ranks.

The SP-BF methods generally perform better than the graph-based SuperGlue model trained on indoor images and the SP-BF-Seg (56.86%) even beats the SP-SGOutdoor-Seg method (55.69%). This indicates that the training data has an effect on the performance of the SuperGlue algorithm. The performance difference between the SGIndoor and SGOutdoor may be due

to the outdoor images resembling the underwater domain to a larger degree than indoor images. However, there is most likely a gap between the terrestrial and underwater domains and we suspect that better performance can be achieved by training a SuperGlue model on underwater images. Even so, the SP-SGOutdoor-BB display the strongest performance with an mAP of 60.34% at full rank.

The results indicate that our solution can significantly reduce the search space for the volunteers who are currently manually matching the images in the Match My Mola project. Instead of comparing with every image in the database, the volunteers may only need to look at the top ranked suggestions to find potential strong matches.

7 Conclusion

We propose a pipeline for re-identification of Giant Sunfish (*Mola alexandrini*) that requires no domain adaptation or training. The pipeline is based on publicly available methods for keypoint detection, description, and matching. The evaluation is based on the novel Tiny Mola Dataset, which consists of underwater images of the Giant Sunfish captured in diverse environments.

We found that the largest impact on performance was based on the choice of descriptor, while the level of segmentation had a low and ambiguous effect. The deep learning based SuperPoint descriptor outperforms the handcrafted keypoint descriptors SIFT and RootSIFT. Good results were obtained with both brute-force matching and the graph neural network based SuperGlue matching. The best performance was achieved using a combination of SuperPoint and SuperGlue with a score of 60.34% mAP on full rank.

None of the methods in the proposed pipeline have been trained or adapted to underwater environments or fish in general. Therefore, the results indicate that the pipeline may also be applicable out-of-the-box in other domains (both terrestrial and underwater). The solution seems especially suited for low-budget or volunteer-based wildlife conservation projects without sufficient data for training supervised machine learning algorithms.

References

[1] E. Sawai, Y. Yamanoue, M. Nyegaard, and Y. Sakai, "Redescription of the bumphead sunfish mola alexandrini (ranzani 1839), senior synonym of mola ramsayi (giglioli 1883), with designation of a neotype for mola mola (linnaeus 1758)(tetraodontiformes: Molidae)," *Ichthyological Research*, vol. 65, no. 1, pp. 142–160, 2018.

[2] M. Nyegaard, "There be giants! the importance of taxonomic clarity of the large ocean sunfishes (genus mola, family molidae) for assessing sunfish vulnerability to anthropogenic pressures." Ph.D. dissertation, Murdoch University, 2018. [Online]. Available: http://researchrepository.murdoch.edu.au/id/eprint/41666

[3] T. Thys, J. P. Ryan, K. C. Weng, M. Erdmann, and J. Tresnati, "Tracking a marine ecotourism star: movements of the short ocean sunfish mola ramsayi in nusa penida, bali, indonesia," *Journal of Marine Biology*, 2016.

[4] Ocean Sunfish Research Trust, "Match my mola," https://oceansunfishresearch.org/matchmymola/, accessed: 2021-09-21.

[5] L. Zheng, Y. Yang, and A. G. Hauptmann, "Person re-identification: Past, present and future," *arXiv preprint arXiv:1610.02984*, 2016. [Online]. Available: https://arxiv.org/abs/1610.02984

[6] Z. Arzoumanian, J. Holmberg, and B. Norman, "An astronomical pattern-matching algorithm for computer-aided identification of whale sharks rhincodon typus," *Journal of Applied Ecology*, vol. 42, no. 6, pp. 999–1011, 2005.

[7] C. W. Speed, M. G. Meekan, and C. J. Bradshaw, "Spot the match–wildlife photo-identification using information theory," *Frontiers in zoology*, vol. 4, no. 1, pp. 1–11, 2007.

[8] O. Moskvyak, F. Maire, F. Dayoub, and M. Baktashmotlagh, "Learning landmark guided embeddings for animal re-identification," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, 2020, pp. 12–19.

[9] J. Bruslund Haurum, A. Karpova, M. Pedersen, S. Hein Bengtson, and T. B. Moeslund, "Re-identification of zebrafish using metric learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, 2020, pp. 1–11.

[10] S. D. McConkey, "Photographic identification of the new zealand sea lion: a new technique," *New Zealand Journal of Marine and Freshwater Research*, 1999.

[11] B. Würsig and T. A. Jefferson, "Methods of photo-identification for small cetaceans," *Reports of the International Whaling Commission. Special*, vol. 12, pp. 42–43, 1990. [Online]. Available: https://www.vliz.be/imisdocs/publications/253951.pdf

[12] A. Van Tienhoven, J. Den Hartog, R. Reijns, and V. Peddemors, "A computer-aided program for pattern-matching of natural marks on the spotted raggedtooth shark carcharias taurus," *Journal of Applied ecology*, vol. 44, no. 2, pp. 273–280, 2007.

[13] J. P. Crall, C. V. Stewart, T. Y. Berger-Wolf, D. I. Rubenstein, and S. R. Sundaresan, "Hotspotter—patterned species instance recognition," in *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE, 2013, pp. 230–237.

[14] W. Wang, R. Solovyev, A. Stempkovsky, D. Telpukhov, and A. Volkov, "Method for whale re-identification based on siamese nets and adversarial training," *Optical Memory and Neural Networks*, vol. 29, no. 2, pp. 118–132, 2020.

[15] E. Nepovinnykh, T. Eerola, and H. Kalviainen, "Siamese network based pelage pattern matching for ringed seal re-identification," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, 2020, pp. 25–34.

[16] S. Bouma, M. D. Pawley, K. Hupman, and A. Gilman, "Individual common dolphin identification via metric embedding learning," in *International conference on image and vision computing New Zealand (IVCNZ)*. IEEE, 2018, pp. 1–6.

[17] O. Moskvyak, F. Maire, F. Dayoub, and M. Baktashmotlagh, "Keypoint-aligned embeddings for image retrieval and re-identification," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2021, pp. 676–685.

[18] S. Schneider, G. W. Taylor, and S. C. Kremer, "Similarity learning networks for animal individual re-identification - beyond the capabilities of a human observer," in *Proceedings of the IEEE Winter Applications of Computer Vision Workshops (WACVW)*, 2020, pp. 44–52.

[19] M. Pedersen, N. Madsen, and T. B. Moeslund, "No machine learning without data: Critical factors to consider when collecting video data in marine environments," vol. 16, no. 3, 2021. [Online]. Available: https://www.thejot.net/archive-issues/?id=73

[20] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2009, pp. 248–255.

[21] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2961–2969.

[22] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[23] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, vol. 2. Ieee, 1999, pp. 1150–1157.

[24] ——, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[25] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 2911–2918.

[26] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 224–236.

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Computer Vision – ECCV 2014*. Cham: Springer, 2014, pp. 740–755.

[28] S. Brin, "Near neighbor search in large metric spaces," in *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, 1995. [Online]. Available: http://ilpubs.stanford.edu:8090/113/

[29] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 4938–4947.

[30] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5828–5839.

[31] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, "Yfcc100m: The new data in multimedia research," *Communications of the ACM*, vol. 59, no. 2, pp. 64–73, 2016.

Paper H

Finding Nemo's Giant Cousin: Keypoint Matching for Robust Re-identification of Giant Sunfish

Malte Pedersen, Marianne Nyegaard, and Thomas B. Moeslund

The paper has been published in the
Journal of Marine Science and Engineering (JMSE), 2023.

The layout has been revised.

Abstract

The Giant Sunfish (Mola alexandrini) has unique patterns on its body, which allow for individual identification. By continuously gathering and matching images, it is possible to monitor and track individuals across location and time. However, matching images manually is a tedious and time-consuming task. To automate the process, we propose a pipeline based on finding and matching keypoints between image pairs. We evaluate our pipeline with four different keypoint descriptors, namely ORB, SIFT, RootSIFT, and SuperPoint, and demonstrate that the number of matching keypoints between a pair of images is a strong indicator for the likelihood that they contain the same individual. The best results are obtained with RootSIFT, which achieves an mAP of 75.91% on our test dataset (TinyMola+) without training or fine-tuning any parts of the pipeline. Furthermore, we show that the pipeline generalizes to other domains, such as re-identification of seals and cows. Lastly, we discuss the impracticality of a ranking-based output for real-life tasks and propose an alternative approach by viewing re-identification as a binary classification. We show that the pipeline can be easily modified with minimal fine-tuning to provide a binary output with a precision of 98% and recall of 44% on the TinyMola+ dataset, which basically eliminates the need for time-consuming manual verification on nearly half the dataset.

1 Introduction

The world's heaviest bony fish, the elusive 'giant sunfish' (*Mola alexandrini*), can reach an impressive weight of more than two tons [1, 2]. Globally, sunfish are rarely seen by divers, but are frequent seasonal visitors to the Bali area, Indonesia [3]. Here, they seek cleaner fish interaction for removal of skin parasites, and are a highly popular target of the local SCUBA tourism industry [4]. Little is known of this seasonal sunfish phenomenon, including if the tourism is reliant on a small, local sunfish population with high site fidelity, or transient sunfish with low re-visitation rates. Understanding this is critical for assessing the potential impacts (and any need for regulation) of diver crowding, which causes disruptions to sunfish–cleaner fish interactions [3].

To investigate this, the citizen science- and volunteer-based project, Match My Mola [5], collects and curates sunfish images from the Bali area, taken by tourist divers, for photo re-identification purposes. Currently, images are manually compared pair-wise, as illustrated in Figure H.1, to assess re-sightings of individuals over time. However, with an increasing number of images, match time becomes a significant challenge. Therefore, we previously proposed a pipeline for automated re-identification of giant sunfish based on keypoint matching, which we published as a workshop paper [6]. We found that our solution worked well as part of a human-in-the-loop system where marine researchers were provided the top-n ranked matches, however, the system relied

on manual labor to visually match and sort out wrong matches. In this article, we expand upon our previous work, providing more insights and proposing an enhanced pipeline that includes and additional pre-processing step to increase the performance. Additionally, we discuss the impracticality of a ranked output and propose a modified pipeline, viewing the re-identification problem as a binary classification, which we argue is more suitable for an efficient human-in-the-loop system.

Our contributions include:

1. A computer-vision-based re-identification pipeline that requires no training or fine-tuning;
2. A demonstration of the generalization attributes of the proposed pipeline;
3. A comparison and performance evaluation of the handcrafted and deep-learning-based keypoint descriptors, i.e., ORB, SIFT, RootSIFT, and SuperPoint, with respect to the re-identification task;
4. A discussion on the impracticality of having a ranked output from re-identification systems and a novel solution to make the proposed system more practical.

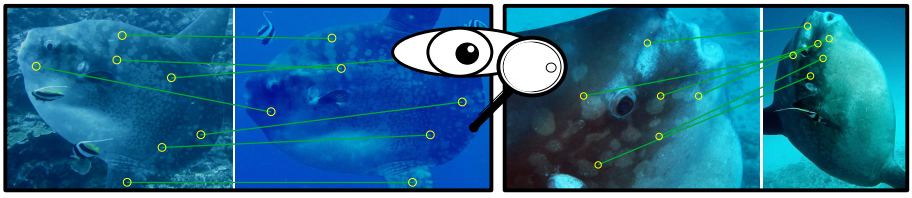


Fig. H.1: Giant sunfish have unique patterns on their bodies, which can be used for photo identification. Traditionally, marine researchers have matched images by manual visual pattern recognition focused on the body markings, as illustrated in these two examples.

2 Related Work

Photographic identification has been used for studying wild animals for decades [7–10]. It allows researchers to identify the same individual across time and location, but it requires manual labor to obtain the photographs and match the individuals from the captured footage. Citizen science projects and camera traps have proven to be effective and irreplaceable methods to gather large amounts of data. However, as a database grows, so does the need for manual labor for identifying the specimens captured in the images or videos. This has lead to an increase in the use of computer vision systems as an assistive tool within biology and ecology [11–13].

Image processing and pattern matching techniques have been used to identify individuals of many types of animals, including whale-sharks [14, 15], spotted raggedtooth sharks [16], fish [17, 18], rays [19], seals [20], birds [21], wild terrestrial animals such as zebras, tigers, polar bears, and giraffes [22–26], and farm animals [27, 28]. Traditional hand-crafted features, such as SIFT [29], RootSIFT [30], and SURF [31], have been used extensively in animal re-identification [22, 27, 32–37]. The common pipeline includes a preprocessing step for finding regions of interest in an image concentrated around the target animal. This is often followed by an image enhancement step aimed at distilling unique patterns on the body of the animal. Keypoints are then detected and features (keypoint descriptors) are extracted centered around these points. This is followed by a matching scheme based on minimizing some distance function between the keypoint descriptors in sets of images [38, 39]. Lastly, the matches are typically cleaned, e.g., using RANSAC [40], to remove potential outliers, and a final set of matches is used to decide on the IDs of the animals based on some similarity score.

Recently, deep-learning-based methods have become a popular means for handling re-identification problems within marine computer vision. Bergler et al. proposed a multi-stage deep-learning-based framework for detecting and identifying individual killer whales [41]. They trained and used a YOLOv3 model [42] for detecting dorsal fins and a multi-class classification ResNet-34 model [43] for determining the identity of the killer whales. Wang et al. proposed to use a Siamese network and adversarial training to identify whales by their flukes [44] and Nepovinnykh et al. trained a Siamese network for Saimaa Ringed Seal re-identification [45]. In another work concerning seals, Chelak et al. [20] proposed a new global pooling technique named EDEN and illustrated that the deep features of a modified and fine-tuned ResNet-18 model are suitable for re-identifying Saimaa Ringed Seals. In the work done by Bouma et al., they trained a ResNet-50 model using a triplet loss for identifying dolphins by their flukes [46]. ResNet-50 was also used by Moskvyak et al. in their work on re-identification of manta rays [47, 48], where they proposed to embed the feature vectors by body landmark information and use a weighted combination of three losses. On a higher level, Schneider et al. investigated how the performance of CNNs was affected by using either Siamese networks or triplet loss for animal re-identification and found that triplet loss generally outperforms Siamese networks [49].

A common property of all the aforementioned methods is the requirement for training data, parameter fine-tuning, or domain adaptation. However, it is demanding to capture images in wild underwater environments and marine image datasets are, therefore, often sparse. This typically leaves little to no room for high-quality training, testing, and validation splits, as is the case with the giant sunfish dataset that we are evaluating in this work.

3 Match My Mola Re-Identification Dataset

Match My Mola is a citizen science- and volunteer-based project that collects images of sunfish from the Bali area in Indonesia. It is the largest curated collection of sunfish images, that we are aware of, and is a valuable resource for ongoing research in the ecology of the giant sunfish in the Bali area. The images in the database are currently only used for photo identification, but the project aims to expand the use of the images in the future to other areas, such as estimating injury rates from local boats and fishing gear. The photo identification approach allows marine scientists to examine if the same individuals frequent the local reefs several times within and between years, and thereby, better understand if the local tourism industry is reliant on residing or transient individuals. In our previous work on re-identification of sunfish [6], we used a subset of the Match My Mola image database, which we named the 'TinyMola' dataset. It consisted of all the (at that time) manually annotated and verified image pairs of the Match My Mola database which totaled 91 images of 29 individuals. However, the Match My Mola image database contains thousands of photos and researchers and volunteers are continuously matching the images by manual visual inspection. This also means that more individuals have been identified since our previous work was conducted and the annotated part of the Match My Mola database currently contains 224 images of 75 individuals. Therefore, to strengthen our findings, we use an expanded second iteration of the TinyMola dataset that contains the images of the 75 individuals. We name this expanded version of the dataset TinyMola+.

Giant sunfish have unique and intricate whitish body patterns which are well suited to identify individuals [50], as has also been suggested for the close relative *Mola mola* [51]. The contrast of the patterns can vary widely depending both on image quality, environmental factors at the time of photography, but also the physiological state of the patterns themselves. Like many other fish species, giant sunfish are capable of rapid physiological coloration change, whereby low-contrast patterns can become bold and clearly visible in seconds [50]. The patterns themselves, however, are stable during the change, and are also stable over at least 7.2 years [50], and are, therefore, a robust characteristic for photo identification.

The images of the Match My Mola database are grouped in photo events (PhE), which contain 1–3 images per side of the same individual captured by the same diver during the same dive. The markings on the giant sunfish are not identical on the two sides, which is also the case for *Mola mola* [51], and they cannot be directly compared, see Figure H.2. Therefore, we frame the re-identification task to be side-specific and provide each side of the fish a unique ID in order to measure the performance of the proposed re-identification pipeline appropriately. For each ID, there are images from at least two PhEs.

In Figure H.2, we present images of a giant sunfish named 'Dabra' from two photo events. Notice that this individual has been recorded from both sides, and therefore, has a unique ID for each side.

Fig. H.2: This giant sunfish has been recorded from both sides in two different photo events (PhE). The patterns on one side cannot be compared to the patterns on the opposite side; therefore, it has a unique ID for each side.

There are cases where two PhEs of the same individual include images from both sides in one of the PhEs but not in the other. With the side-specific composition, TinyMola+ has a total of 224 images containing 83 IDs divided into 41 left-sided and 42 right-sided IDs, with 116 and 108 images, respectively. The images have been gathered from a total of 166 photo events. Most of the images are captured by tourist divers and the quality of the images varies extensively. This is amplified by the turbidity of the water, attenuation of light, occlusion, image compression, and more [52]. Examples illustrating some of the variation of the dataset can be seen in Figure H.3. The resolution of the original images varies from 0.1 to 16 megapixels (MP) with a mean around 4 MP and an object resolution around 1 MP on average. However, all the images from the Match My Mola database have been manually cropped around the sunfish by the researchers who curate the database to ease their manual inspections. We use the cropped images in our work to focus on the re-identification task.

Fig. H.3: Example images from TinyMola+. The quality varies widely between the images and some have high resolutions, clear objects, and distinct patterns, while others have very low resolutions, reduced contrast, and vague patterns.

4 Method

The process of identifying the sunfish on the images is currently conducted manually by marine researchers or trained volunteers. This includes cropping the image around the target and comparing markings across all overlapping body parts on the two images. The images are compared pair-wise and matches are noted and examined by other matching experts, to confirm that the images are of the same individual. Our solution, described below and illustrated in Figure H.4, is inspired by the manual process and is an improvement to our previous work on the re-identification of the giant sunfish [6], with an additional pre-processing step.

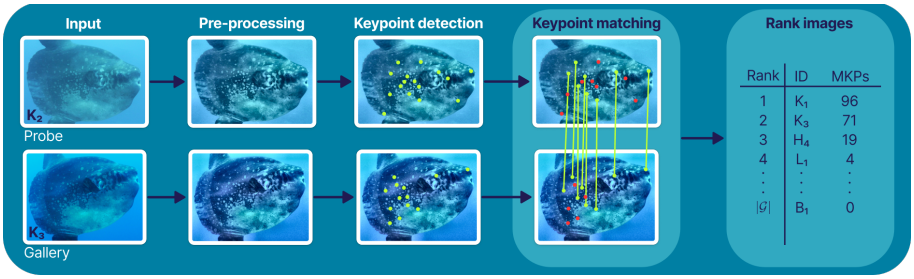


Fig. H.4: The proposed module-based pipeline with illustrations of an image pair containing the same individual with ID = K (the subscripted number signifies that the images are not the same). In the pre-processing module, the image contrast is enhanced for both images and then the keypoints are detected and matched in the following two modules. Lastly, the gallery images are ranked based on the number of matching keypoints (MKPs), where a higher number indicates a stronger similarity.

4.1 Pre-Processing—Contrast Enhancement

The quality of the images in the TinyMola+ dataset varies to a large degree depending on when and where the images were taken and by whom. This leads to the patterns on the sunfish being less pronounced in some images, e.g., due to the low contrast or low quality of the image. For this reason, we investigated how contrast enhancement may affect the performance of the pipeline. The aim was not to create realistic out-of-the-water images of the sunfish, but rather to enhance the clarity of the patterns to potentially allow for an increased number of distinct keypoints. It should be noted that this module is an addition to our previously proposed method [6].

For enhancing the contrast, we chose the well-proven contrast limited adaptive histogram equalization method [53] (CLAHE). CLAHE enhances the contrast in the image adaptively by processing the image as a set of smaller patches and equalizing the local patch-based histograms (in opposition to a global

equalization, which often leads to undesirable results). In addition to the adaptive equalization, CLAHE includes a clipping step to minimize the enhancement of noise. We present a range of varied examples from TinyMola+ processed by CLAHE in Figure H.5.

Fig. H.5: The images in the first row have not been processed. The second row contains the same images after the contrast has been enhanced using the CLAHE algorithm. Notice how the patterns stand out more clearly as the contrast is increased.

4.2 Keypoint Detection

The body of a sunfish is highly rigid, except for the dorsal and anal fins [54]. Consequently, sunfish captured on images at different times and locations are mainly affected by affine transformations, such as rotation and scale, and we can utilize this to determine whether a pair of images contains the same individual. Detecting and describing points affected by such transformations have been studied intensively for decades in fields such as image registration and tracking. Candidate locations are typically known as keypoints or interest points and they must be characteristic in some manner, e.g., a corner or a high intensity pixel in a low intensity neighborhood. In this work, we evaluated the performance of three handcrafted feature descriptors, i.e., SIFT [29, 55], RootSIFT [30], and ORB [56], and the state of the art deep-learning-based feature descriptor SuperPoint [57] with respect to the re-identification of individual sunfish. Each of the aforementioned methods are summarized in this section.

Probably the most widely used hand-crafted keypoint descriptor is the Scale Invariant Feature Transform (SIFT) [29, 55], which was proposed two decades ago. SIFT features are based on extrema points that are consistent throughout a difference of Gaussians scale space. When an extrema point has been located, a histogram of oriented gradients (HoG) is created from the region surrounding the pixel. An orientation is assigned to the keypoint based on the normalized HoG features. The SIFT keypoint descriptor itself is based on a 4×4 matrix of normalized HoG features with 8 bins, resulting in a feature vector with 128 values.

Following the publication of SIFT, Arandjelovic and Zissermann noted that matching the features using Euclidean distance, as proposed in the original pa-

per, is not always the best solution [30]. SIFT features are based on histograms and the Hellinger kernel is typically preferred over Euclidean distance when comparing histograms. Therefore, Arandjelovic and Zissermann proposed to L1-normalize the SIFT feature vector and take the square root of each element subsequently, naming this new feature descriptor RootSIFT. Practically, this means that matching RootSIFT features using the Euclidean distance is equivalent to matching SIFT features using the Hellinger kernel, which typically improves the results.

A more recent hand-crafted feature descriptor is the Oriented FAST and Rotated BRIEF [56] (ORB), which, as the name implies, is based on a combination of the FAST keypoint detector [58] and BRIEF keypoint descriptor [59]. FAST and BRIEF were designed to be both fast and accurate, but have several downsides. FAST responds greatly along edges, but has no way of measuring cornerness, which is otherwise typically considered a salient feature. In ORB, this is handled by sorting the keypoints using a Harris corner measure [60] and only picking the top candidates. Furthermore, a scale pyramid is employed and features are produced at each level to handle the lack of scale-invariant keypoints. Lastly, a keypoint orientation is included based on the assumption that the intensity of corners are offset from their centers. This also solves the main problem with the BRIEF descriptor not being able to handle rotations. The orientation of the FAST corner keypoint is used to steer the BRIEF descriptor, which thereby becomes rotation-invariant.

Learned feature descriptors have gained attention with the popularization and accessibility of strong GPUs to efficiently train deep learning models. A seminal work in this field is the self-supervised keypoint detector and descriptor SuperPoint [57], which is a fully convolutional neural network (CNN) with a shared encoder and two decoder heads for keypoint detection and description, respectively. The authors proposed to train a base model with just the decoder head for keypoint detection, named MagicPoint, on purely synthetic data of angular-shaped objects. They then used MagicPoint to create pseudo ground-truth labels on real images. This was done by warping the input images using random homographic transformations, detecting keypoints in the warped images, and aggregating the unwarped set of keypoints into a superset of labels. The homographic adaption of real images allows for jointly self-supervised training of the SuperPoint keypoint detector and descriptor to be invariant to scaling and rotation.

Giant sunfish re-identification presents challenges to all four descriptors. For example, Figure H.6 shows two image pairs of the same individual along with the matching keypoints (MKPs). The first example contains two images with relatively low contrast, while the individuals in the second example are rotated in relation to each other. These examples highlight some of the obstacles that can complicate keypoint detection and matching for the respective algorithms.

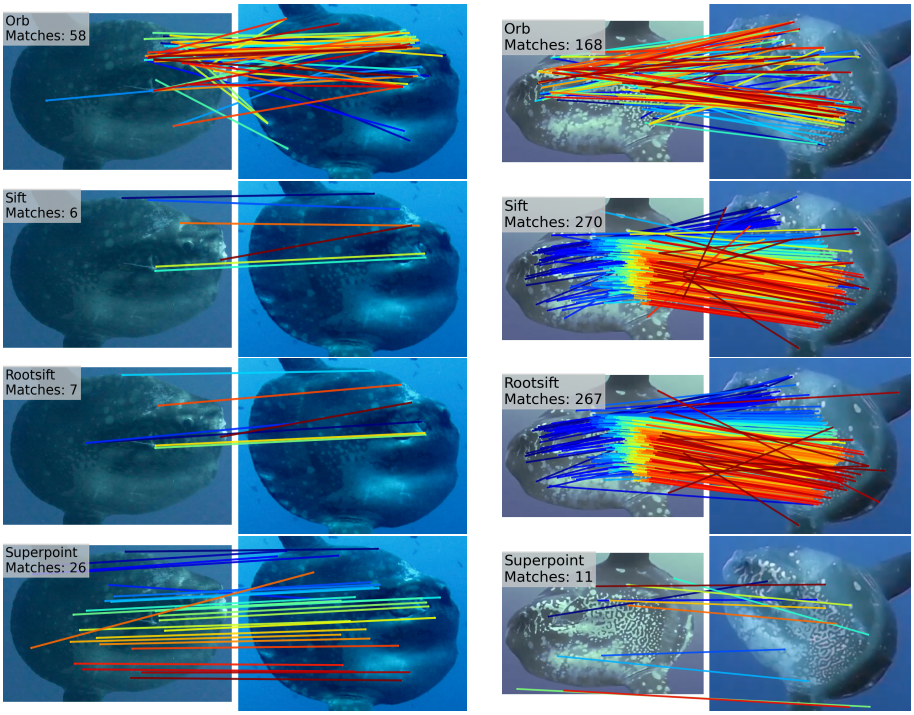


Fig. H.6: Two examples of image pairs with certain characteristics that can complicate the process of detecting and matching keypoints. The image pairs in the first example (left) have relatively low contrast, while the image pairs in the second example (right) are heavily rotated in relation to each other. In the first example, ORB finds a multitude of matching keypoints, though a large part of them are false positives, SIFT and RootSIFT find few and relatively imprecise keypoints, and lastly, SuperPoint finds a decent amount of mostly precise keypoints. In the second example, ORB finds both correct and incorrect matching keypoints, SIFT and RootSIFT find many correct matches, and SuperPoint finds very few matches. Note that the images have not been contrast-enhanced and the colors of the lines are only for visualization.

4.3 Keypoint Matching

The keypoints are described by feature vectors, and to determine whether keypoints in two images represent the same point on the object, we measured the distance between the vectors. Depending on the problem, dimensionality, and nature of the data, keypoint matching has commonly been performed using brute-force methods or kd-trees [61]. Brute-force methods compare all elements in the two distributions and are guaranteed to find the best match, but the processing time can be high for large distributions. On the other hand, kd-trees do not guarantee to find the best match, but are faster for large distributions. As there are, in our case, no time constraints on the task and the dataset is relatively small, we performed an exhaustive search and matched the

keypoints using a brute-force method. SIFT, RootSIFT, and SuperPoint features were matched based on the L^2 distance and the ORB features were matched based on the Hamming distance due to the binary nature of the features.

Naively matching the closest keypoints can lead to poor results. For this reason, David G. Lowe introduced the distance ratio test [29] as a way to dismiss keypoints that are ambiguous. If the ratio between the distance to the nearest and second nearest neighbor is above a threshold, the keypoint is considered too uncertain and is discarded. The optimal threshold depends on the nature of the data, and if it is too low, too many correct matches may be discarded and vice versa. We used the distance ratio test as the last step of the keypoint matching module to clean our matches.

4.4 Ranking Images

For every image pair, we viewed the number of matching keypoints as a similarity score, where a higher number of MKPs indicates a stronger similarity. We sorted and ranked all images based on the number of MKPs, as illustrated in Figure H.4. Note that the example in the figure is hypothetical and only for visualization purposes.

5 Evaluation Protocol

It is not possible to manually determine whether the left and right side of a giant sunfish belong to the same individual, except where photos exist of both sides during the same photo event. Therefore, each side of an individual was assigned different IDs. In cases where a photo event contains images of both sides, but only one of the sides has a match from another photo event in the dataset, the unmatched image was named a *single* and was considered to be noise. Every image, except singles, was considered a *probe* $p \in \mathcal{P}$. Each probe was compared against the *gallery* images, with $g \in \mathcal{G}$ being the set of all images in the dataset except the probe. Note that the singles were included in the gallery and there was always at least one gallery image with the same ID as the probe.

5.1 Performance Metrics

Re-identification systems are typically evaluated based on their ability to rank the gallery images by their similarity to the probe. Two of the most commonly used metrics for evaluating ranking-based re-identification systems are the cumulative matching characteristic (CMC) [62] and the mean average precision (mAP) [63]. The CMC describes the accuracy of the system at a given rank and is often presented as rank-k accuracy. The CMC score is inadequate in cases

where the gallery contains multiple images that ID with the probe, as it only refers to the highest ranked true positive gallery sample. Therefore, we also evaluated our system by the mAP, which punishes suboptimal ordering of the ranked gallery images. The CMC score at rank x was computed as follows:

$$\mathrm{CMC}^x = \frac{1}{|\mathcal{P}|}\sum_{p}^{\mathcal{P}} \begin{cases} 1, & \text{if any of the top-}x\text{ ranked gallery images shares ID with } p \\ 0, & \text{otherwise} \end{cases} \tag{H.1}$$

We calculated the average precision for probe p at rank x as follows:

$$\mathrm{AP}_p^x = \frac{1}{\mathcal{H}_p}\sum_{n=1}^{x} \mathrm{pr}_n R_n \tag{H.2}$$

where $\mathcal{H}_p = \min\{|g_p|, x\}$, $|g_p|$ is the total number of gallery images that shares ID with the probe, and R is a relevance function given by:

$$R = \begin{cases} 1, & \text{if the gallery image is a true positive} \\ 0, & \text{otherwise} \end{cases} \tag{H.3}$$

Moreover, pr is the precision, calculated as follows:

$$\mathrm{pr} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{H.4}$$

where TP is the number of true positives and FP is the number of false positives. Finally, the mAP for rank x was found by:

$$\mathrm{mAP}^x = \frac{1}{|\mathcal{P}|}\sum_{p}^{\mathcal{P}} \mathrm{AP}_p^x \tag{H.5}$$

5.2 Pipeline Parameters

An essential aspect of this work was to design a pipeline that is suited for non-technical staff. Therefore, in order to minimize the need for user involvement, we chose default parameters for the methods included in the pipeline. In the pre-processing module, the image contrast was enhanced with CLAHE. We did not fine-tune the CLAHE parameters, but used a patch-size of 8×8 pixels and a clipping limit of 3, which are commonly used settings. We used the default settings for the four keypoint descriptors and we cleaned the matches of SIFT, RootSIFT, and SuperPoint using the distance ratio test with a default threshold of 0.8, as proposed by Lowe in the original SIFT paper [29]. We generally used Python as the programming language and the OpenCV library for implementing the image processing algorithms, such as CLAHE, SIFT, RootSIFT, and ORB. We used the pre-trained SuperPoint model from Magic Leap [57] implemented in PyTorch.

5.3 Testing Data

Beside evaluating our pipeline on the TinyMola+ dataset, we included two additional re-identification datasets with patterned animals for a more thorough assessment of the system, namely, the $\text{SealID}_{\text{patches}}$ [20] and OpenCows2020 [28] datasets. As can be seen from the examples in Figure H.7, the three datasets vary widely with respect to object appearance, image quality, and contrast, as well as the number of images per individual.

- The TinyMola+ dataset contains 83 individuals and 224 images of varying sizes.
- The $\text{SealID}_{\text{patches}}$ test split contains 26 individuals and 836 images of size 256×256.
- The OpenCows2020 test split contains 46 individuals and 496 images of varying sizes.

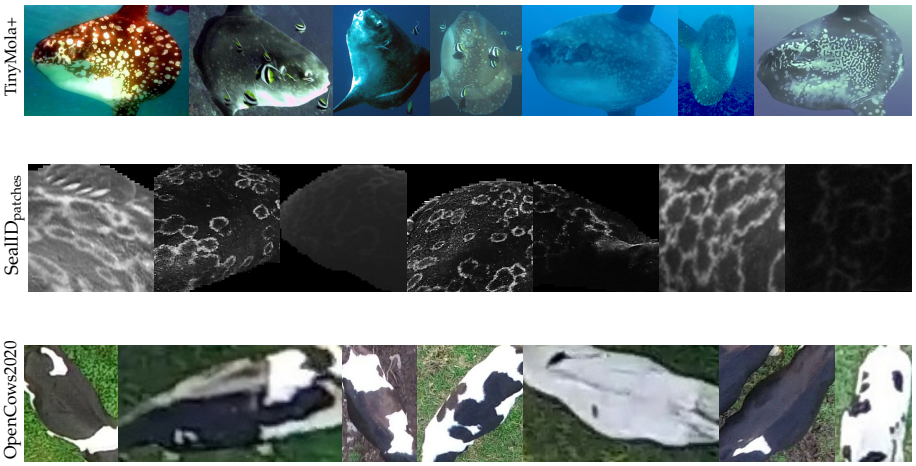


Fig. H.7: Examples from the TinyMola+, $\text{SealID}_{\text{patches}}$ [20], and OpenCows2020 [28] datasets.

During evaluation, all images of the TinyMola+ and OpenCows2020 datasets were resized to a maximum dimension of 640 pixels while keeping the aspect ratio to ensure that the size of the objects were approximately similar between the images. We did not resize the images of the $\text{SealID}_{\text{patches}}$ dataset, as they had already been resized to 256×256 by the authors of the dataset in order to ensure that the patterns are of approximately the same scale. Note that we exclusively evaluated on the testing splits; we did not use the training splits, as we did not train nor fine-tune our pipeline.

6 Results

Recall that the aim of this work was to develop an automated re-identification pipeline that requires no training data, as it can be extremely difficult and time consuming to capture sufficient data to train robust supervised models for marine tasks due to the harsh underwater environment. Furthermore, underwater environments can vary widely visually, which means that the pipeline should be able to generalize well. Hence, we conducted an evaluation of the efficiency and adaptability of the suggested pipeline equipped with each of the keypoint descriptors, in order to determine the optimal choice among the four candidates.

First, we demonstrate the superiority of the new proposed pipeline compared to the former pipeline [6] on the TinyMola+ dataset. Hereafter, we show that the pipeline generalizes to other re-identification subjects with distinct patterns by evaluating the system on two very different datasets: $\text{SealID}_{\text{patches}}$ [20] and OpenCows2020 [28]. In Table H.1, we present results from our former and new pipeline on the TinyMola+ dataset. We see a tendency indicating that the number of matching keypoints can serve as a strong predictor for determining whether two images contain the same individual. Our new pipeline outperforms the former solution with the SIFT, RootSIFT, and SuperPoint descriptors; however, it performs significantly worse with ORB. The descriptor that obtains the best performance on the TinyMola+ dataset is RootSIFT, which reaches an mAP of 75.91%.

Table H.1: Results from the former [6] and current pipeline on the TinyMola+ dataset. We present the CMC score for three ranks (1, 3, and 5). The best results are highlighted in bold. The difference between the former and current solution is highlighted in green if the current solution is better and red otherwise.

	TinyMola+			
Model	CMC^1	CMC^3	CMC^5	mAP
$\text{Former}_{SuperPoint}$	69.20	72.32	75.00	60.74
Ours_{ORB}	29.02 (−40.18)	36.16 (−36.16)	39.73 (−35.27)	23.97 (−36.77)
Ours_{SIFT}	76.79 (+7.59)	82.14 (+9.82)	83.04 (+8.04)	70.20 (+9.46)
$\text{Ours}_{RootSIFT}$	**80.36** (+11.16)	**84.38** (+12.06)	**86.16** (+11.16)	**75.91** (+15.17)
$\text{Ours}_{SuperPoint}$	72.32 (+ 3.12)	77.23 (+ 4.91)	77.68 (+ 2.68)	63.88 (+ 3.14)

We present an overview of the results from the $\text{SealID}_{\text{patches}}$ and OpenCows2020 datasets in Table H.2. On $\text{SealID}_{\text{patches}}$, our pipeline outperforms the deep-learning-based and supervised solution proposed by the authors of the dataset [20] with respect to the rank-1 accuracy. This is not the case for the OpenCows2020 dataset, where the authors present a pre-trained ResNet-50 model fine-tuned with a combination of a softmax and reciprocal triplet loss

(RTL) [28] that we are not able to match, although we obtained reasonable results. It should be noted that it is unclear exactly how the authors of both datasets calculate their accuracy, but to the best of our knowledge, it is the CMC rank-1 accuracy. Additionally, we evaluated our method on the entire testing split for both datasets and did not consider portions of known/unknown ID's between the testing and training splits as we did not need to train nor fine-tune our pipeline as opposed to the other two solutions that needed annotated training data.

We see that RootSIFT has a marginally higher mAP compared to SuperPoint on the $\text{SealID}_{\text{patches}}$ dataset, while the opposite is true for the CMC rank-1 score. This indicates that the two descriptors basically perform equally well on the dataset and they are closely followed by SIFT. It is another story when looking at the results for the OpenCows2020 dataset. SIFT and RootSIFT get extremely low CMC rank-1 scores and the mAPs are also substantially lower compared to SuperPoint. This is possibly due to CLAHE not having the desired effect on the OpenCows2020 dataset, which contains a multitude of very small images. A suboptimal configuration of CLAHE may induce an enhancement of noisy elements instead of the actual patterns on the cow (which already have a high contrast due to their naturally black and white colorization).

Table H.2: Results from the $\text{SealID}_{\text{patches}}$ [20] and OpenCows2020 [28] datasets. We present the CMC score for three ranks (1, 3, and 5). The best results are highlighted in bold. Note that our pipeline runs off-the-shelf, while the other solutions are trained specifically for the task at hand.

	$\text{SealID}_{\text{patches}}$				**OpenCows2020**			
Model	CMC^1	CMC^3	CMC^5	mAP	CMC^1	CMC^3	CMC^5	mAP
EDEN [20]	86.54	-	-	-	-	-	-	-
$\text{ResNet50}_{\text{Softmax-RTL}}$ [28]	-	-	-	-	**87.55**	-	-	-
Ours_{ORB}	77.87	83.49	86.24	31.70	35.08	43.55	50.60	22.59
Ours_{SIFT}	92.82	95.93	96.29	49.95	0.81	77.02	83.67	28.75
$\text{Ours}_{RootSIFT}$	93.18	**97.01**	**97.49**	**57.34**	0.81	82.46	**86.69**	31.51
$\text{Ours}_{SuperPoint}$	**94.86**	96.89	97.13	56.97	73.79	**83.27**	85.69	**39.52**

We present three examples from the OpenCows2020 and $\text{SealID}_{\text{patches}}$ datasets in Figures H.8 and H.9 for each of the descriptors, respectively. All the images in the examples have been processed by CLAHE. We see a tendency that ORB generally finds many, but unreliable, matching keypoints. SIFT and RootSIFT find many true MKPs, but also a portion of false MKPs between images of the same individual. SuperPoint generally finds fewer MKPs compared to the other descriptors, but they seem to be more robust and very few false positives are found.

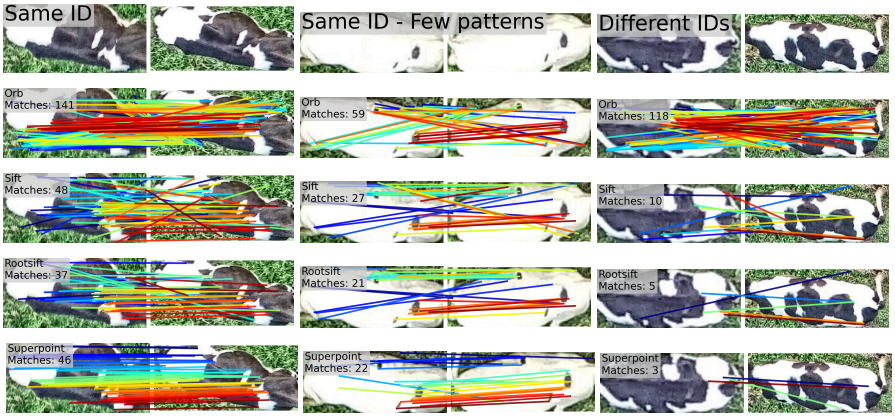


Fig. H.8: Matching examples from the OpenCows2020 dataset. The images have been processed by CLAHE. The colors of the lines are only for visualization.

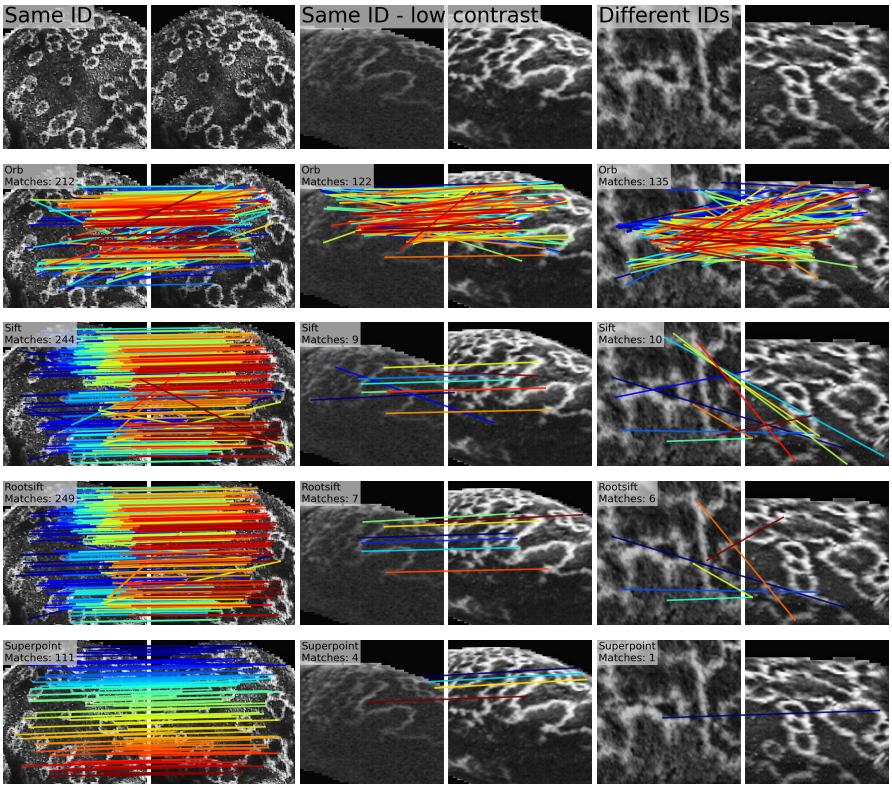


Fig. H.9: Matching examples from the $\text{SealID}_{\text{patches}}$ dataset. The images have been processed by CLAHE. The colors of the lines are only for visualization.

7 Discussion

We have seen that our proposed pipeline performs well on the TinyMola+ dataset and also seems to generalize well to other similar tasks. However, conducting meaningful research on wild and elusive animals such as the giant sunfish based on re-identification is challenging due to the time-consuming task of obtaining and analyzing sufficient amounts of data. Therefore, it is not uncommon that volunteers assist in data collection and data curation in environmental and conservation projects. However, this also entails that the personnel on these projects typically have diverse backgrounds and it cannot be expected that they have technical skills to configure or train complicated computer vision systems. Beside proposing a system that works in practice out-of-the-box, an important part of this work is to design a system that requires absolute minimal intervention from the user. A common approach to solve re-identification tasks is by providing the top-n ranked images based on some similarity score, as we did above. However, a ranked output has some negative application-specific attributes:

1. It is not obvious how to decide the optimal rank;
2. It is time consuming to manually verify matches (both positive and negative);
3. It is difficult to evaluate the practicality of the system by standard metrics, such as mAP and CMC.

In real-life applications, there is typically a human in the loop that needs to verify the output of the re-identification system. Often, the user needs to decide on the number of gallery images to look through (the rank). If the rank is too low, the user will miss positive samples, and if the rank is too high, the user will have to look through a multitude of false positives.

In short, a ranking-based output is not very practical in real-life applications. Alternatively, we suggest that the re-identification task can be viewed as a binary classification problem, where a pair of images can either contain the same individual or not. This allows for an arbitrary number of gallery images that share an ID with the probe while liberating the user from deciding on the number of proposals per probe to look through. In the following section, we present a novel binary classification module as an alternative to the ranking module and discuss its strengths and weaknesses.

7.1 Re-Identification as a Binary Classification Problem

Only minor adjustments are required for the pipeline to deliver a binary output. One method is to accept every image pair that has at least a single pair of matching keypoints as a positive sample. However, as we know that all the keypoint descriptors are likely to find noisy MKPs, this will lead to a huge number of disordered false positive identifications, which is even less practical than the ranked output. The key to a robust binary classification module is a very high precision, meaning that very few false positives are accepted.

Thresholding the Minimum Number of Matching Keypoints

A way to minimize the number of false positives is to find a threshold for the minimum number of MKPs needed for an image pair to be considered a positive match. However, such a threshold includes finding a compromise between reducing the number of false positives and increasing the number of false negatives. This compromise can be visualized through a precision–recall curve, where the precision and recall are calculated as:

$$\mathrm{pr} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \qquad \mathrm{rc} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{H.6}$$

where TP are true positives, FP are false positives, and FN are false negatives. A fabricated example with a probe and five gallery images can be seen in Figure H.10. In the given example, one image has been correctly matched (a TP) and three others have been wrongly matched (three FPs), illustrated by the green check mark and red crosses, respectively. The right-most image has the same ID as the probe, but has not been matched (an FN). In the given example, pr = 0.25 and rc = 0.5.

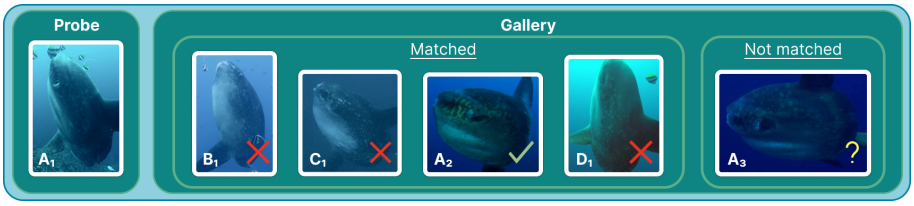


Fig. H.10: An example with a probe and a gallery containing five images. The ID of each fish is presented in the bottom left corner of the image by a letter. Four of the gallery images have been matched with the probe; however, only the gallery image with the green check mark is correctly matched. The gallery image that has not been matched with the probe shares the ID of the probe. In total, this gives three false positives, one true positive, and one false negative.

The optimal compromise between precision and recall depends on the task at hand. In our case, precision is critical and we want the highest possible precision in order to remove the need for manual labor of verifying the samples.

We present two precision–recall plots for the TinyMola+ dataset in Figure H.11 based on varying the minimum number of MKPs. Both plots contain curves for each of the four keypoint descriptors.

The left plot contains a traditional precision–recall curve that shows that SuperPoint reaches the best compromise between precision and recall, which is highlighted by the area under the curve (AUC) presented in the legends. The circles refer to the tipping point of the precision recall-curve's 'shoulder', where the precision starts to decrease. The plot on the right is an elaboration of the precision and recall values with respect to the MKPs threshold. Precision and recall are both visualized on the vertical axis, while the horizontal axis indicates the minimum number of MKPs. Note that the circles in the two plots mark the same precision and recall values.

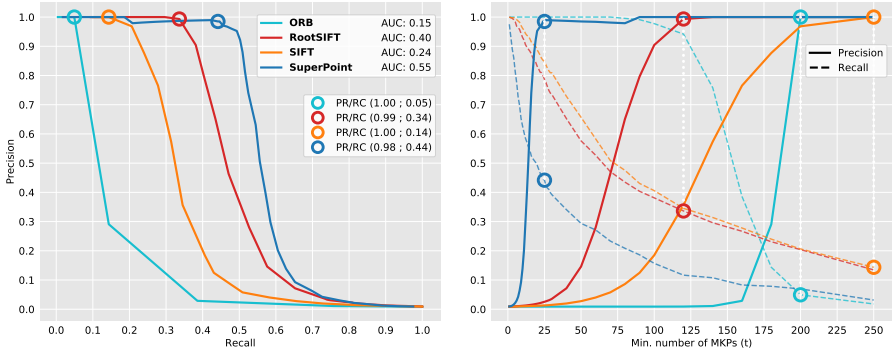


Fig. H.11: The left plot shows a precision–recall curve based on varying the minimum number of MKPs needed for an image pair to be classified as a positive match. The right plot expands upon the precision–recall curve by showing both precision and recall plotted on the vertical axis and the minimum number of matching keypoints on the horizontal axis. The circles in both plots highlight the precision and recall values at the 'shoulder' of the precision–recall curve.

ORB and SIFT give the worst performance with high precision at the expense of very low recall. RootSIFT outperforms the two other handcrafted descriptors by reaching a recall of 0.34 and a precision of 0.99 at $t = 120$. Lastly, we observe that SuperPoint is able to reach a recall of 0.44 and precision of 0.98 at $t = 25$. This means that 0.44% of the matches of the TinyMola+ dataset are found with a very high precision among the image pairs that shares at least 25 MKPs, and it basically removes the need for user involvement in almost half of the dataset. However, the number of MKPs alone is not the only parameter that we can tune to remove false positives. By analyzing the composition of the matching keypoints, we may be able to allow fewer MKPs, and thereby, a higher recall, while preserving a high precision.

Thresholding the Maximum Condition Number

The body of a sunfish is nearly flat and completely rigid, meaning that it approximates a plane well. We can exploit this by assuming that the matching keypoints between an image pair point have the same spatial positions on two identical planes, which allows us to compute the homography and estimate the change in rotation and distance between the images. In our case, the homography describes the projective transformation between the planes spanned by the bodies of the sunfish, which is naturally constrained. In the odd cases where we have matching keypoints between images of *different* individuals, the projective transformation between the planes will be unconstrained and ambiguous. We can utilize this to minimize the number of false positive matches by discarding image pairs with unlikely projective transformations.

A way to determine the unlikeliness of a projective transformation is by looking at the condition number of the homography matrix. The condition number, κ, indicates to what degree a change in the input affects the output. In the case of a homography matrix, this means that if the matrix is based on a range of correct MKPs, a limited projective transformation is produced, after which a small change to the input will only cause a small change to the output. However, if the matches between the image pairs are wrong, they will point to random spatial positions on the planes and not agree on a common transformation. This leads to a system where even small changes to the input will significantly alter the output. We calculate the condition number as follows:

$$\kappa(H) = \frac{\sigma_{\max}(H)}{\sigma_{\min}(H)} \tag{H.7}$$

where $\sigma_{\max}$ and $\sigma_{\min}$ are the maximum and minimum singular values, respectively, and H is the homography matrix computed from the matching keypoints. A minimum of four matching keypoints are needed in order to estimate the homography [64], but more MKPs are preferred to minimize the impact of noisy matches. The condition number lies in the interval $[1, +\infty]$, where a lower number implies a stronger candidate for a correct match and a higher number indicates a more complex and unlikely transformation.

The examples presented in Figure H.12 are visual illustrations of the correlation between the condition number of the homography matrix and the soundness of the output image. Note that SIFT are used in all the examples; however, similar patterns are observed for the other descriptors. The first example contains an image pair of the same individual captured from different angles. The MKPs are largely correct, and this is illustrated by a relatively simple projective transformation that causes the first image to align nicely with the second image. The following four examples contain different individuals, meaning that all MKPs are wrong. This is illustrated by complicated projective transformations that lead to absurd and unrecognizable projected images.

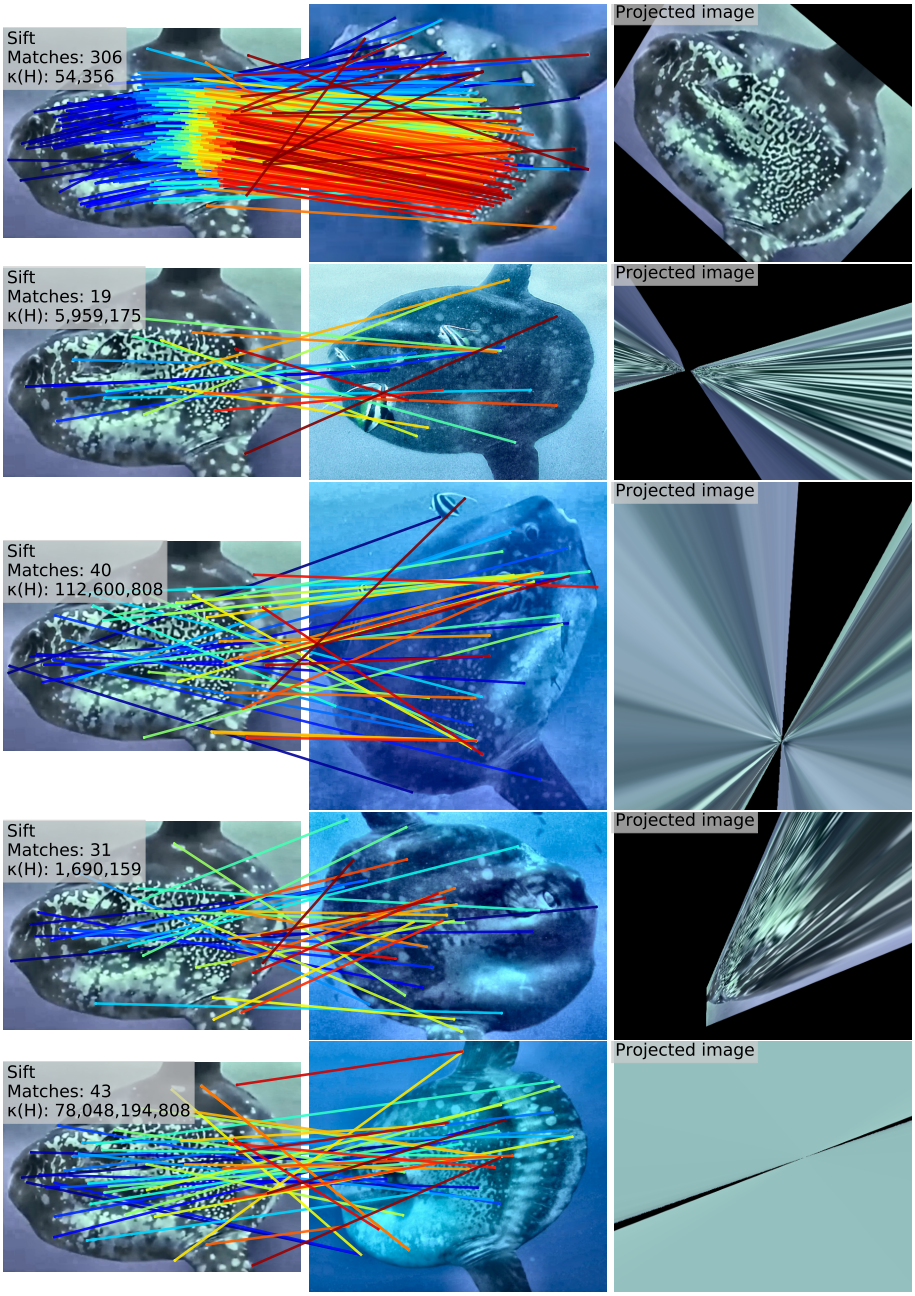


Fig. H.12: The example in the first row shows an image pair of the same individual and the projected image in the third column is well aligned with the second image, as expected. The following four rows show image pairs that contain different individuals, which means that the matching keypoints are wrong per definition. The consequence is odd homography matrices that lead to absurd projective transformations, as illustrated by the projected images in the third column.

7.2 Evaluating the Binary Classification

In Figure H.13, we present plots with precision–recall curves for each of the keypoint descriptors when thresholding both the minimum number of MKPs and the maximum condition number. Each curve in the plot is based on varying the threshold, t, for the minimum number of MKPs, while the color of the curve indicates the value of the condition number threshold parameter $\mathcal{L}$. The four highlighted curves resemble the curves presented in Figure H.11, with no threshold on the condition number. We see that the performance of SIFT, RootSIFT, and ORB can be significantly improved by thresholding both parameters, whereas the gain is negligible for SuperPoint. This is highlighted by the AUC presented in the legends, which is sorted based on the threshold value $\mathcal{L}$.

The best performance, according to the AUC, is obtained when $\mathcal{L} \approx 1 \times 10^5$, while lower AUC scores are seen for both higher and lower threshold values. However, SuperPoint is an exception to the latter as it reaches an optimal and stable performance for $\mathcal{L} \geq 1 \times 10^6$. This indicates that the SuperPoint features are more robust compared to the other descriptors and there is no substantial gain in looking at the condition number, as every image pair that has at least 25 MKPs is practically certain to be a correct match. It is possible to obtain comparable results with SIFT and RootSIFT with respect to the AUC, but it requires the user to fine-tune the minimum number of keypoints as well as the maximum condition number, making them less applicable compared to SuperPoint.

7.3 Summary

We demonstrate a practical alternative to the typical supervised ranking-based re-identification model with the proposed pipeline equipped with SuperPoint and the novel binary classification module. The main drawback of this approach is the need for a (minimal) fine-tuning of the MKPs threshold parameter. However, the binary output has a range of benefits:

- It allows for an arbitrary number of gallery images that share an ID with the probe (without forcing the user to be concerned about choosing an optimal rank for the proposals);
- It effectively reduces the need for human verification;
- It allows for training supervised models on the automatically labeled data (essentially making them unsupervised);
- The binary and ranked output can be combined by removing the binary classified positive samples and only manually inspecting the top-n proposals of the remainder of the dataset.

Although thresholding the condition number did not result in increased performance for SuperPoint, the use of homography to transform images allows for a practical method of evaluating ranked proposals. Our examples shown in Figure H.12 indicate that image pairs depicting different individuals display obscure projective transformations, making it easy to distinguish projected images of matching vs. non-matching individuals. Adding transformed images as an additional source to the original images may enhance the practicality of our pipeline for manual verification of the top-n ranked images, even though its effect cannot be quantified by traditional metrics, such as mAP, CMC, precision, or recall. Future research on the re-identification of giant sunfish, and re-identification in general, should involve real-life assessments that consider the entire human-in-the-loop setup to evaluate the efficacy of different strategies in terms of accuracy, practicality, and efficiency. In particular, the benefits of ranking vs. binary classification should be evaluated.

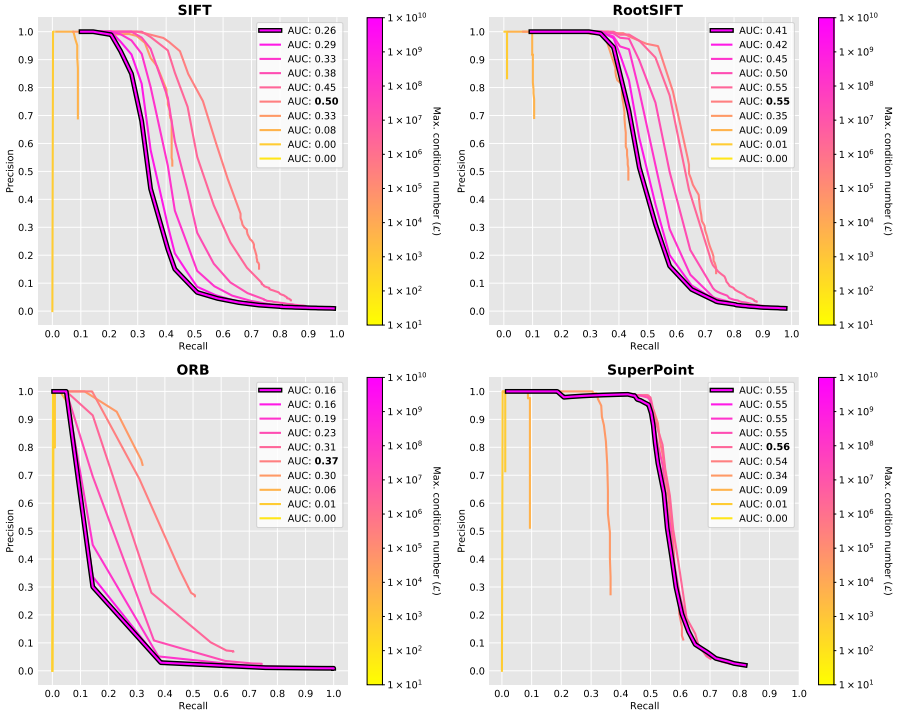


Fig. H.13: Precision–recall curves for each of the four keypoint descriptors. The curves are based on varying the minimum number of matching keypoints from 1 to 250, while the colors indicate the threshold of the condition number. The legends present the AUC values sorted by the condition number threshold parameter $\mathcal{L}$. Note that thresholding the condition number is nonessential for SuperPoint, as the performance is stable when $\mathcal{L} \geq 1 \times 10^6$.

8 Conclusions

We propose a computer-vision-based pipeline for identifying individual giant sunfish using keypoint matching. We evaluate the pipeline equipped with each of the four keypoint descriptors: ORB, SIFT, RootSIFT, and SuperPoint. The pipeline achieved a mean average precision of 75.91% on the TinyMola+ dataset without any training or fine-tuning. Furthermore, we demonstrate that the pipeline generalizes well to other patterned species, such as seals and cattle, where its performance is comparable to state-of-the-art supervised methods concerning the CMC rank-1 score.

Lastly, we argue that a ranking-based output is not practical for real-life scenarios, as it is challenging for users to determine an optimal rank. Instead, we consider the re-identification task as a binary classification and introduce an alternative output module that identifies image pairs with at least a single pair of matching keypoints as positives. Initially, this approach resulted in a high number of false positives, making it impractical. However, by only accepting image pairs with at least 25 matching keypoints, we demonstrate that giant sunfish can be robustly identified with a precision of 98%, a recall of 44%, and an area under the precision–recall curve of 55%. This approach eliminates the need for human verification of almost half of the TinyMola+ dataset.

Further research is required to thoroughly investigate how automated computer-vision-based re-identification systems can be integrated into practical human-in-the-loop systems. A carefully considered balance between automated and human decision making is required to ensure that such systems are effective and efficient in real-life scenarios and not just on the drawing board.

References

[1] J. N. Gomes-Pereira, C. K. Pham, J. Miodonski, M. A. R. Santos, G. Dionísio, D. Catarino, M. Nyegaard, E. Sawai, G. P. Carreira, and P. Afonso, "The heaviest bony fish in the world: A 2744-kg giant sunfish mola alexandrini (ranzani, 1839) from the north atlantic," *Journal of Fish Biology*, vol. 102, no. 1, pp. 290–293, nov 2022.

[2] E. Sawai and M. Nyegaard, "A review of giants: Examining the species identities of the world's heaviest extant bony fishes (ocean sunfishes, family molidae)," *Journal of Fish Biology*, vol. 100, no. 6, pp. 1345–1364, apr 2022.

[3] M. Nyegaard, "There be giants! the importance of taxonomic clarity of the large ocean sunfishes (genus mola, family molidae) for assessing sunfish vulnerability to anthropogenic pressures." Ph.D. dissertation, Murdoch University, 2018. [Online]. Available: http://researchrepository.murdoch.edu.au/id/eprint/41666

[4] T. Thys, J. P. Ryan, K. C. Weng, M. Erdmann, and J. Tresnati, "Tracking a marine ecotourism star: movements of the short ocean sunfish mola ramsayi in nusa penida, bali, indonesia," *Journal of Marine Biology*, 2016.

[5] Ocean Sunfish Research Trust, "Match my mola," https://oceansunfishresearch.org/matchmymola/, accessed: 2021-09-21.

[6] M. Pedersen, J. B. Haurum, T. B. Moeslund, and M. Nyegaard, "Re-identification of giant sunfish using keypoint matching," vol. 3, 3 2022.

[7] S. Schneider, G. W. Taylor, S. Linquist, and S. C. Kremer, "Past, present and future approaches using computer vision for animal re-identification from camera trap data," *Methods in Ecology and Evolution*, vol. 10, no. 4, pp. 461–470, jan 2019.

[8] P. S. Hammond, S. A. Mizroch, and G. P. Donovan, *Individual recognition of cetaceans: use of photo-identification and other techniques to estimate population parameters: incorporating the proceedings of the symposium and workshop on individual recognition and the estimation of cetacean population parameters*. International Whaling Commission, 1990.

[9] S. D. McConkey, "Photographic identification of the new zealand sea lion: a new technique," *New Zealand Journal of Marine and Freshwater Research*, 1999.

[10] B. Würsig and T. A. Jefferson, "Methods of photo-identification for small cetaceans," *Reports of the International Whaling Commission. Special*, vol. 12, pp. 42–43, 1990. [Online]. Available: https://www.vliz.be/imisdocs/publications/253951.pdf

[11] B. G. Weinstein, "A computer vision for animal ecology," *Journal of Animal Ecology*, vol. 87, no. 3, pp. 533–545, nov 2017.

[12] N. Petrellis, "Measurement of fish morphological features through image processing and deep learning techniques," *Applied Sciences*, vol. 11, no. 10, 2021.

[13] M. Goodwin, K. T. Halvorsen, L. Jiao, K. M. Knausgård, A. H. Martin, M. Moyano, R. A. Oomen, J. H. Rasmussen, T. K. Sørdalen, and S. H. Thorbjørnsen, "Unlocking the potential of deep learning for marine ecology: overview, applications, and outlook," *ICES Journal of Marine Science*, vol. 79, no. 2, pp. 319–336, Jan 2022.

[14] Z. Arzoumanian, J. Holmberg, and B. Norman, "An astronomical pattern-matching algorithm for computer-aided identification of whale sharks rhincodon typus," *Journal of Applied Ecology*, vol. 42, no. 6, pp. 999–1011, 2005.

[15] J. Holmberg, B. Norman, and Z. Arzoumanian, "Estimating population size, structure, and residency time for whale sharks rhincodon typus through collaborative photo-identification," *Endangered Species Research*, vol. 7, pp. 39–53, apr 2009.

[16] A. Van Tienhoven, J. Den Hartog, R. Reijns, and V. Peddemors, "A computer-aided program for pattern-matching of natural marks on the spotted raggedtooth shark carcharias taurus," *Journal of Applied ecology*, vol. 44, no. 2, pp. 273–280, 2007.

[17] J. Bruslund Haurum, A. Karpova, M. Pedersen, S. Hein Bengtson, and T. B. Moeslund, "Re-identification of zebrafish using metric learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, 2020, pp. 1–11.

[18] Ø. L. Olsen, T. K. Sørdalen, M. Goodwin, K. Malde, K. M. Knausgård, and K. T. Halvorsen, "A contrastive learning approach for individual re-identification in a wild fish population," vol. 4, Jan. 2023.

[19] N. Gómez-Vargas, A. Alonso-Fernández, R. Blanquero, and L. T. Antelo, "Re-identification of fish individuals of undulate skate via deep learning within a few-shot context," *Ecological Informatics*, vol. 75, p. 102036, 2023.

[20] I. Chelak, E. Nepovinnykh, T. Eerola, H. Kälviäinen, and I. Belykh, "EDEN: Deep feature distribution pooling for saimaa ringed seals pattern matching," in *Cyber-Physical Systems and Control II*. Springer International Publishing, 2023, pp. 141–150.

[21] A. C. Ferreira, L. R. Silva, F. Renna, H. B. Brandl, J. P. Renoult, D. R. Farine, R. Covas, and C. Doutrelant, "Deep learning-based methods for individual recognition in small birds," *Methods in Ecology and Evolution*, vol. 11, no. 9, pp. 1072–1085, jul 2020.

[22] J. P. Crall, C. V. Stewart, T. Y. Berger-Wolf, D. I. Rubenstein, and S. R. Sundaresan, "Hotspotter—patterned species instance recognition," in *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE, 2013, pp. 230–237.

[23] A. Shukla, C. Anderson, G. S. Cheema, P. Gao, S. Onda, D. Anshumaan, S. Anand, and R. Farrell, "A hybrid approach to tiger re-identification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, oct 2019.

[24] C. W. Speed, M. G. Meekan, and C. J. Bradshaw, "Spot the match–wildlife photo-identification using information theory," *Frontiers in zoology*, vol. 4, no. 1, pp. 1–11, 2007.

[25] C. J. R. Anderson, N. D. V. Lobo, J. D. Roth, and J. M. Waterman, "Computer-aided photo-identification system with an application to polar bears based on whisker spot patterns," *Journal of Mammalogy*, vol. 91, no. 6, pp. 1350–1359, dec 2010.

[26] J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. Rubenstein, "Animal population censusing at scale with citizen science and photographic identification," in *SS-17-01*, ser. AAAI Spring Symposium - Technical Report. United States: AI

Access Foundation, 2017, pp. 37–44, publisher Copyright: © Copyright 2017, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.; 2017 AAAI Spring Symposium ; Conference date: 27-03-2017 Through 29-03-2017.

[27] W. Andrew, S. Hannuna, N. Campbell, and T. Burghardt, "Automatic individual holstein friesian cattle identification via selective local coat pattern matching in RGB-d imagery," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, sep 2016.

[28] W. Andrew, J. Gao, S. Mullan, N. Campbell, A. W. Dowsey, and T. Burghardt, "Visual identification of individual holstein-friesian cattle via deep metric learning," *Computers and Electronics in Agriculture*, vol. 185, p. 106133, jun 2021.

[29] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[30] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 2911–2918.

[31] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Computer Vision – ECCV 2006*. Springer, 2006, pp. 404–417.

[32] R. Maglietta, V. Renò, G. Cipriano, C. Fanizza, A. Milella, E. Stella, and R. Carlucci, "DolFin: an innovative digital platform for studying risso's dolphins in the northern ionian sea (north-eastern central mediterranean)," *Scientific Reports*, vol. 8, no. 1, nov 2018.

[33] L. Zhao, M. Pedersen, J. Y. Hardeberg, and B. Dervo, "Image-based recognition of individual trouts in the wild," in *Proceedings of the European Workshop on Visual Information Processing (EUVIP)*. IEEE, 2019.

[34] S.-L. Long and N. A. Azmi, "Using photographic identification to monitor sea turtle populations at perhentian islands marine park in malaysia," *Herpetological Conservation and Biology*, vol. 12, no. 2, pp. 350–366, 2017.

[35] M. C. Stoddard, R. M. Kilner, and C. Town, "Pattern recognition algorithm reveals how birds evolve individual egg pattern signatures," *Nature Communications*, vol. 5, no. 1, jun 2014.

[36] S. G. Dunbar, E. C. Anger, J. R. Parham, C. Kingen, M. K. Wright, C. T. Hayes, S. Safi, J. Holmberg, L. Salinas, and D. S. Baumbach, "HotSpotter: Using a computer-driven photo-id application to identify sea turtles," *Journal of Experimental Marine Biology and Ecology*, vol. 535, p. 151490, feb 2021.

[37] D. T. Bolger, T. A. Morrison, B. Vance, D. Lee, and H. Farid, "A computer-assisted system for photographic mark-recapture analysis," *Methods in Ecology and Evolution*, vol. 3, no. 5, pp. 813–822, may 2012.

[38] A. Moghimi, T. Celik, A. Mohammadzadeh, and H. Kusetogullari, "Comparison of keypoint detectors and descriptors for relative radiometric normalization of bitemporal remote sensing images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 4063–4073, 2021.

[39] S. A. K. Tareen and Z. Saleem, "A comparative analysis of sift, surf, kaze, akaze, orb, and brisk," in *Proceedings of the International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2018, pp. 1–10.

[40] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[41] C. Bergler, A. Gebhard, J. R. Towers, L. Butyrev, G. J. Sutton, T. J. H. Shaw, A. Maier, and E. Nöth, "FIN-PRINT a fully-automated multi-stage deep-learning-based framework for the individual recognition of killer whales," *Scientific Reports*, vol. 11, no. 1, dec 2021.

[42] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv:1804.02767*, 2018.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016, pp. 770–778.

[44] W. Wang, R. Solovyev, A. Stempkovsky, D. Telpukhov, and A. Volkov, "Method for whale re-identification based on siamese nets and adversarial training," *Optical Memory and Neural Networks*, vol. 29, no. 2, pp. 118–132, 2020.

[45] E. Nepovinnykh, T. Eerola, and H. Kalviainen, "Siamese network based pelage pattern matching for ringed seal re-identification," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, 2020, pp. 25–34.

[46] S. Bouma, M. D. Pawley, K. Hupman, and A. Gilman, "Individual common dolphin identification via metric embedding learning," in *International conference on image and vision computing New Zealand (IVCNZ)*. IEEE, 2018, pp. 1–6.

[47] O. Moskvyak, F. Maire, F. Dayoub, and M. Baktashmotlagh, "Learning landmark guided embeddings for animal re-identification," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, 2020, pp. 12–19.

[48] ——, "Keypoint-aligned embeddings for image retrieval and re-identification," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2021, pp. 676–685.

[49] S. Schneider, G. W. Taylor, and S. C. Kremer, "Similarity learning networks for animal individual re-identification - beyond the capabilities of a human observer," in *Proceedings of the IEEE Winter Applications of Computer Vision Workshops (WACVW)*, 2020, pp. 44–52.

[50] M. Nyegaard, J. Karmy, L. McBride, T. M. Thys, M. Welly, and R. Djohani, "Rapid physiological colouration change is a challenge - but not a hindrance - to successful photo identification of giant sunfish (mola alexandrini, molidae)," *Frontiers in Marine Science*, vol. 10, may 2023.

[51] T. Kushimoto, A. Kakino, and N. Shimomura, "Possible individual identifications by the body surface marking patterns in the ocean sunfish *Mola mola* and the sharptail sunfish *Masturus lanceolatus* (molidae)," *Ichthy, Natural History of Fishes of Japan*, vol. 19, pp. 1–7, 2022.

[52] M. Pedersen, N. Madsen, and T. B. Moeslund, "No machine learning without data: Critical factors to consider when collecting video data in marine environments," vol. 16, no. 3, 2021. [Online]. Available: https://www.thejot.net/archive-issues/?id=73

[53] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive histogram equalization and its variations," *Computer Vision, Graphics, and Image Processing*, vol. 39, no. 3, pp. 355–368, September 1987.

[54] Y. Watanabe and K. Sato, "Functional dorsoventral symmetry in relation to lift-based swimming in the ocean sunfish mola mola," *PLoS ONE*, vol. 3, no. 10, p. e3446, oct 2008.

[55] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, vol. 2. Ieee, 1999, pp. 1150–1157.

[56] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Ieee. IEEE, November 2011, pp. 2564–2571.

[57] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 224–236.

[58] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32, no. 1, pp. 105–119, January 2010.

[59] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Computer Vision – ECCV 2010*. Springer, 2010, pp. 778–792.

[60] C. Harris and M. Stephens, "A combined corner and edge detector," *Alvey Vision Conference*, pp. 147–151, 1988.

[61] S. Brin, "Near neighbor search in large metric spaces," in *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, 1995. [Online]. Available: http://ilpubs.stanford.edu:8090/113/

[62] H. Moon and P. J. Phillips, "Computational and performance aspects of PCA-based face-recognition algorithms," *Perception*, vol. 30, no. 3, pp. 303–321, mar 2001.

[63] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, dec 2015.

[64] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, mar 2004.

ISSN (online): 2446-1628
ISBN (online): 978-87-7573-693-5

AALBORG UNIVERSITY PRESS