FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Automated Analysis of Markerless Freestyle Swimming Metrics

Sara Gabriela Almeida Marinha

Mestrado em Engenharia Informática e Computação

Supervisors: Prof. Pedro C. Diniz and Prof. J. Paulo Vilas-Boas

July 25, 2024

Automated Analysis of Markerless Freestyle Swimming Metrics

Sara Gabriela Almeida Marinha

Mestrado em Engenharia Informática e Computação

July 25, 2024

Resumo

Melhorar o desempenho de um atleta e prevenir lesões depende fortemente da análise do seu movimento. No caso dos nadadores, registar os seus movimentos é crucial para avaliar e melhorar várias métricas, como a velocidade de natação, frequência de braçadas, durações das fases subaquática e aérea, frequência da pernada, profundidade do movimento e ângulos das articulações.

Tradicionalmente, marcadores têm sido usados para enquadrar automaticamente nadadores em vídeos, mas este método é dispendioso e demorado. Além disso, é intrusivo, pois a presença física dos marcadores pode por vezes impedir o desempenho de um nadador. Reconhecendo estas limitações, tem havido um impulso na direção da utilização de técnicas sem marcadores, ou *markerless*. Estas tecnologias aproveitam câmaras de vídeo convencionais, reduzindo complexidades de configuração e custos. Algoritmos avançados de identificação de movimento aceleram ainda mais a aquisição de dados, minimizando a entrada subjetiva do operador na identificação de pontos anatómicos chave. No entanto, estes métodos mais recentes enfrentam desafios devido aos efeitos da água — como distorção da água e bolhas — tornando o processo notavelmente difícil.

O objetivo principal desta dissertação consiste no desenvolvimento de um conjunto de algoritmos de processamento de imagem adaptados para análise de movimento de nadadores usando técnicas de vídeo sem marcadores. Esta abordagem inovadora visa superar as complexidades impostas por ambientes aquáticos, utilizando um sistema composto por uma câmara móvel. A abordagem centra-se em técnicas de estimação de pontos anatómicos através da análise de *frames* de vídeo, visando extrair métricas de desempenho essenciais como frequência e profundidade de braçadas.

Esta pesquisa visa acelerar a análise de desempenho de nadadores, com o desenvolvimento de ferramentas para avaliar vários aspectos da técnica de natação do atleta *markerless*, divergindo das técnicas atuais trabalhosas e potencialmente intrusivas.

Abstract

Enhancing an athlete's performance and preventing injuries hinges greatly on analyzing their actions. In the case of swimmers, recording their movements is pivotal for assessing and improving various metrics such as swimming speed, stroke frequency, duration of underwater and aerial phases, kicking frequency, depth of movement, and joint angles.

Traditionally, markers have been employed to automatically track swimmers in videos, but this method is expensive and time-consuming. Moreover, it can sometimes impair swimmer's performance. Recognizing these limitations, there's been a push towards markerless techniques as a promising avenue for motion analysis, especially in aquatic settings. These technologies leverage conventional video cameras, reducing setup complexities and costs. Advanced motion identification algorithms further expedite data acquisition by minimizing subjective operator input in identifying key anatomical points. However, these newer methods face challenges due to the water's effects — such as water distortion and bubbles — making image processing notably difficult.

The primary objective of this dissertation is the development of a suite of image processing algorithms tailored for swimmer motion analysis using markerless video techniques. This innovative approach aims to overcome the complexities posed by aquatic environments, utilizing a system comprising one mobile camera. The approach centers around employing anatomical point estimation techniques via video frame analysis, aiming to extract essential performance metrics like average displacement speed and stroke frequency.

This research endeavors to fasten swimmer performance analysis by providing tools to evaluate various aspects of the athlete's swimming technique without relying on markers, diverging from the current laborious and potentially intrusive techniques.

Acknowledgments

This stage of my life would have been much more difficult without the support and help of many people and organizations. I am incredibly grateful to all of you.

First, I want to express my heartfelt thanks to my supervisor, Professor Pedro C. Diniz, for your invaluable guidance, patience, and encouragement. Your advice and expertise were crucial to the completion of this work.

I am also grateful to my friends and colleagues, who supported me every step of the way.

A huge thank you to my family and my boyfriend for their unwavering love and belief in me. Your support has been my rock throughout this process.

Finally, I extend my gratitude to the Faculty of Sport at the University of Porto for providing the resources for my research. A special thank you to Professor J. Paulo Vilas-Boas for inspiring the initial idea for this project.

Thank you all!

Sara Marinha

"It always seems impossible until it's done."

Nelson Mandela

Contents

List of Figures

List of Tables

Symbols and Abbreviations

2D	Two Dimensional
3D	Three Dimensional
AI	Artificial Intelligence
AVI	Audio Video Interleave
CNN	Convolutional Neural Networks
FPS	Frames Per Second
GPU	Graphics Processing Unit
IMU	Inertial Measurement Unit
JSON	JavaScript Object Notation
LSTM	Long- Short-Term Memory
ML	Machine Learning
RQ	Research Question
SMPL	skinned Multi-Person Linear model

Chapter 1

Introduction

1.1 Context and Motivation

Swimming is a complex sport that involves the coordination of movements in a fluid medium, where both the underwater and aerial phases play crucial roles in performance. Traditional methods of analyzing swimming techniques often involve manual observation and qualitative assessments, which can be subjective and lack precision or in other instances use physical markers which not only are time-intensive and costly but also intrusive, potentially impeding the natural performance of the swimmer. Moreover, capturing the full scope of the movement of the swimmer requires synchronization between multiple cameras, both above and underwater, complicating the analysis process. There is a clear need for a systematic, accurate, and comprehensive method to analyze swimming techniques to help swimmers optimize their performance and coaches to provide more targeted training strategies.

In the context of high-performance swimming sports, the uncovering and subsequent detailed analysis of specific metrics are crucial for enhancing performance. By obtaining specific detailed movement metrics, both the coach and the swimmer, can gain valuable insights into the performance of the swimmer, highlighting areas of strength thus pinpointing opportunities for further improvement.

Accurate extraction of swimming movement data is thus critically important. However, current state-of-the-art data capturing approaches make use of body markers for accurate pinpointing of selected anatomic locations of the athlete, such as the heaps, knee, and arms. While these provide accurate location that are both potentially very intrusive and can fall off or be dislodged in particular in environments such as water with high hydrodynamic parameters.

This work explores an alternative *markerless* approach. Rather than relying on the intrusive body markers, we make use of image processing analysis techniques to automatically detect selected anatomic body points and by using existing anatomic models, we derive a characterization of the movements of the athlete.

Over the past years, *markerless* technology has attracted an increasing attention for its potential to make motion analysis accessible. This technology enables the use of conventional video

cameras, thus reducing costs and the complexity of experimental setups. It employs motion identification algorithms, in particular based on anatomic models, that reduce operator subjectivity in pinpointing anatomical points, thus speeding up the results and increasing the accuracy of the acquisition process.

In this context, the `OpenPose` framework has been used for the analysis of land-based sports videos such as basketball and tennis. Still, *markerless* solutions using `OpenPose` remain underexplored for aquatic sports, particularly swimming. This is not surprising as aquatic environment pose unique, and serious, challenges such as the water-air interface, refractive effects, air bubbles, and the oscillation of skin masses, necessitate specialized solutions. Furthermore, the positioning of a swimmer relative to the camera plane, in a sagital position, creates unique anatomic recognition challenges not present in other sports. These challenges become exacerbated in three-dimensional analyses.

This work, therefore, aims at evaluating the suitability of a *markerless* method based on an open-source video analysis and modeling software package - the `OpenPose` (7) to automatically derive movement metrics for swimmers using video recordings using a single low-cost digital camera. To the best of our knowledge, this is the first work using these off-the-shelf solutions for swimming.

1.2 Objectives

This work aims to address the various challenges outlined above by developing a suite of digital image processing algorithms that enable the analysis of swimming-relevant parameters using *markerless* video techniques, applied in a system with a single camera. The objective is to develop and evaluate a prototype toolbox that includes several stages, each addressing a specific challenge, to achieve the overarching goal of enhancing swimmer analysis. This toolbox is envisioned to simplify the process of capturing and analyzing movements of swimmers, making it more efficient and less intrusive than existing marker-based methods, thereby offering an innovative approach to swimmer performance analysis in aquatic environments.

1.3 Hypothesis and Research Questions

This dissertation considers the following hypothesis:

Is it possible the use `OpenPose` *for the accurate derivation of key performance metrics in aquatic sports, such as in free-style swimming?*

Under the assumption that this hypothesis is *true*, in a set of practical contexts to be determined, a subsequent set of research questions (RQ) which we aim at addressing include:

- **RQ1**: Can the *OpenPose* models retrieve a statistically significant number of anatomic keypoints of interest of the swimmer's position that correlate with relevant performance metrics?
- **RQ2**:In particular, is the use of a single commercially off-the-shelf camera (of reasonable image quality) a low-cost solution with acceptable cost-benefit?
- **RQ3**: Is the approach too computationally expensive to be deemed practical?
- **RQ4**: Is the accuracy of the derived performance metrics comparable to the ones derived by manual inspection of the video recordings of the sports activity?

Although we strongly believe that answer to these questions is positive, to the best of our knowledge the use of `OpenPose` toolkit library has not been extensively used evaluated in the context of the analysis of underwater video images for swimming with a low-cost solution.

1.4 Document Structure

This chapter provides the context and motivation for the study, outlines its objectives, presents the thesis, and formulates the research questions. It sets the stage for the problem being addressed and underscores the significance of the research.

Chapter 2 reviews the existing literature and previous work related to underwater swimming analysis and *markerless* motion capture techniques. It highlights the current state of the art and identifies the gaps this research aims to fill.

Chapter 3 details the implementation and methodology developed to extract swimming metrics. It provides a comprehensive description of the toolbox and the digital image processing algorithms employed.

Chapter 4 presents the results of data analysis performed on the output of `OpenPose` and the developed algorithms. It includes a thorough discussion of these results, evaluating the effectiveness and accuracy of the proposed methods.

Finally, Chapter 5 summarizes the main findings, revisits the research questions, and outlines future work aimed at enhancing the toolbox and extending its application.

Chapter 2

State of the Art

This chapter provides an in-depth review of the current state-of-the-art techniques and technologies in motion capture, image capture, movement estimation, and metrics extraction, with a particular focus on their applications in the context of sports evaluation. The discussion encompasses both traditional and modern methods, highlighting their approaches, advantages, and limitations.

2.1 Motion Capture

Motion capture (MoCap) systems are sophisticated technologies designed to capture and analyze human motion. These systems are broadly categorized into optical and non-optical types, each with unique techniques and applications across various fields such as entertainment, sports, and medical rehabilitation (5).

2.1.1 Techniques

2.1.1.1 Optical Motion Capture

Optical motion capture techniques can be further divided into marker-based and *markerless* systems.

- **Marker-Based systems**: These systems can be either passive or active, based on the type of markers used.

  Passive systems utilize retro-reflective markers that reflect light back to cameras equipped with sensors. Examples include the Ariel system, Motion Analysis' HiRes system, Peak Performance's Motus system, Qualisys' ProReflex system, BTS's ElitePlus system, and Vicon's 370 system. These systems are known for their rapid data collection capabilities but may require post-processing to handle issues such as marker dropout (24).

  Active optical systems, such as the CODA system, use markers with built-in light sources. These systems capture 3D data immediately without extensive tracking or editing procedures, enhancing real-time data acquisition (24).

- **Markerless Systems**: Systems, such as Theia3D, capture motion without physical markers by using multiple synchronized video cameras and advanced image processing algorithms. These systems are less restrictive regarding subject attire and data collection environment, offering significant advantages in terms of setup time and subject comfort (9).

2.1.1.2 Non-Optical Motion Capture

Non-optical techniques include inertial, magnetic, and mechanical systems.

- **Inertial**: Inertial Measurement Unit (IMU)-based systems employ accelerometers, gyroscopes, magnetometers, and signal transmission chips to capture motion. They are more affordable and portable compared to optical systems, making them suitable for various environments, including clinical and home settings. However, IMUs are prone to sensor drift, leading to inaccuracies in long-duration recordings, and require accurate calibration for reliable measurements. Sophisticated algorithms are needed to integrate data from multiple IMUs to provide a coherent picture of body motion (5).
- **Magnetic**: Magnetic-based systems, such as Skill Technology's 6D Research system, use magnetic sensors to determine segment positions and orientations. These systems face significant challenges in aquatic environments due to the distortion and refraction of light in water, which can impede the accurate tracking of markers (24).
- **Mechanical**: Mechanical systems traditionally involve the use of markers and infrared cameras to record and analyze whole-body motion. These systems, while comprehensive, are less commonly used in contemporary applications due to advancements in optical and inertial technologies.

2.1.2 Application to Sports

Motion capture technology has revolutionized various aspects of sports, providing significant advancements in physical education, referee assistance, technological action innovation, body indicator capture, and virtual reality training. It enables detailed analysis of athletes' movement patterns, aiding in injury prevention, performance enhancement, and the development of tailored training programs (6). Markerless systems, in particular, offer greater flexibility and ease of use in naturalistic settings, allowing for more accurate and comprehensive biomechanical assessments. For instance, the use of Theia3D has shown that markerless systems can capture kinematic and kinetic data with minimal effect from subjects' running attire, making them suitable for real-world sports applications (9).

2.2 Image Capture and Basic Analysis

The fields of image capture and motion capture are closely intertwined, especially in applications requiring precise analysis of dynamic objects and human motion. Motion capture technologies,

such as optical and non-optical systems, play a crucial role in capturing and analyzing human movement in various environments, from sports arenas to clinical settings.

In image capture and basic analysis, the focus often lies in identifying regions of interest within static or dynamic scenes. For dynamic camera setups, the challenges include uncertainties related to camera movement and occlusion by other moving objects (22; 30). These challenges are akin to those faced in motion capture, where precise tracking of markers or subjects in motion is essential for accurate data collection and analysis.

To address uncertainties regarding the positioning of objects of interest, the common practice involves the use of Kalman filtering techniques (21). While extensively researched, particularly in contexts of robotics (movement of the robots and its manipulators), these methods are often computationally demanding and suitable only for non-critical settings in terms of power, weight, and non-real time settings (10). Irrespective, of the primary image analysis, at the end of this phase a set of potential objects of interest and the corresponding motion estimation analysis is obtained.

In the domain of aquatic image analysis, the calibration of underwater cameras is a critical aspect. The solution proposed by Yangzhou (26) leverages vanishing point optimization based on two orthogonal parallel lines. This rapid calibration technique tackles image distortion challenges arising from air-glass-water light refraction, impacting both accuracy and speed in monocular underwater camera calibration. Through a series of steps, including correction of distortion using a glass-water-based light refraction model and precise construction of vanishing points, this approach swiftly derives essential camera parameters.

2.3 Movement Estimation and Modeling

After this phase, approaches collect early trajectory estimation and refine them, to compose and determine the consistency of movements from which the final metrics or interest are derived. Classical approaches include the use of basic center-of-mass identification and linear regression techniques for the modeling of trajectories. More recent approaches use Convolution Neural Networks (CNNs) (27) for the projection of trajectories and thus with supervised learning reduce the need for computationally intensive steps. Still other recent approaches leverage computer vision algorithms to estimate the human pose without relying on markers and in the context of swimming, using only a single 8 Mpixel wide-angle camera placed below the surface of the water (4).

In other contexts, and in particular in very controlled in-door environments, researchers have focused on the use of open-source software such as `Pose2Sim` (18) for the modeling of specific movements of human bodies possibly even in 3D. The software, positioned at the intersection of computer vision and bio-mechanics, heavily relies on `OpenPose` for identifying 2D key-points. Its adaptability becomes evident in its customizable procedures, which include fine-tuning the camera, accurately tracking subjects in 2D, robustly triangulating 3D key-points, and meticulously filtering 3D coordinates. These steps converge to produce a detailed skeletal in a `OpenSim` model that comprehensively represents the movements of the subject.

One common issue on movement estimation is occlusion, which some researchers tackled by separating occluder and occludee characteristics within the feature space. Their approach involves incorporating spatial regularization, channel attention mechanisms, and a multi-task learning framework. These elements collaborate to facilitate the regression of 3D body model parameters, leveraging occlusion-aware features for enhanced accuracy (22). Shenming Feng and Haifeng Hu (3) proposed an alternative method to address this issue. Their approach employs two subnets: one focuses on extracting joint structures using a multi-branch feature extraction approach and pyramid residual units, while the other subnet, employing coordinate regression and an attention mechanism, fine-tunes the spatial relationships between joints. Beyond tackling occlusion, their method also effectively addresses additional challenges in human pose estimation, including diverse poses and back views.

Bas Van Hooren et al. (8) explores the differences between the computer vision techniques, namely DeepLabCut and `OpenPose`, in conjunction with markerless motion capture for evaluating different sagittal-plane in running kinematics at two different speeds. It revealed that while `OpenPose` exhibited similarities to marker-based methods, DeepLabCut showcased more significant disparities. Given the variance in accuracy among individuals, `OpenPose` appears more suitable for comprehensive data collection and analyzing group trends rather than individual-level assessments.

While these options are open-source, there are also commercial solutions, such as `Theia3D` and `Captury`, which offer more refined and possibly more user-friendly alternatives for specific applications. Tishya et al. (29) have examined these commercial systems in detail, comparing their performance and reliability to both open-source solutions and traditional marker-based systems.

`Theia3D`, in particular, is designed for clinical applications and has been evaluated for its ability to perform reliable gait analysis in children with various gait abnormalities. It eliminates the need for physical markers, which reduces setup time and allows for a more natural range of motion during data collection. Theia3D uses deep learning algorithms to identify and track anatomical landmarks, generating detailed 3D kinematic data from standard video recordings (29).

However, commercial systems like `Theia3D` also have their challenges. They can be expensive and may require specialized hardware and software setups. Additionally, because these systems are proprietary, users have limited ability to customize the algorithms or the data processing workflow. Despite these challenges, commercial systems are often preferred in clinical settings due to their ease of use, robust customer support, and validated performance metrics (29).

A major advantage of *markerless* motion capture, whether open-source or commercial, is the reduction in inter-assessor variability. Traditional marker-based systems require significant expertise to place markers accurately, and even slight differences in placement can lead to variations in the data. *Markerless* systems, by contrast, use consistent algorithms to identify anatomical landmarks, which improves the reliability of the data across different sessions and assessors (29).

In conclusion, *markerless* motion capture shows great promise as an alternative to traditional marker-based systems for gait analysis and other movement studies. Systems like `Theia3D` and `Captury` offer convenient, time-efficient solutions with the potential for wide application in both

clinical and research settings. However, further validation and research are needed to address specific challenges, such as tracking in the presence of occlusions and ensuring accuracy for complex movements and anatomies. As technology advances and more data become available for training these algorithms, the performance of *markerless* systems is expected to improve, making them an increasingly viable option for detailed motion analysis (29).

`OpenPose` has been widely employed across various land-based sports, including tennis (25), volleyball, soccer (19), squash, karate, boxing, and basketball (13). Its applications range from predicting shot directions and estimating player movements to detecting body orientations and key postures, as well as analyzing athletic performance and techniques. For instance, Shimizu et al. (25) developed a method using `OpenPose` to predict shot directions in tennis matches by leveraging pose information and player position, utilizing Long- Short-term Memory (LSTM) models for prediction. Nakai et al. (13) utilized `OpenPose` to create a posture analysis model for predicting the shooting probability of basketball free throws, highlighting OpenPose's practicality and cost-effectiveness in motion analysis. Moreover, Fritz (19) explored `OpenPose`'s utility in analyzing body orientation during penalty kicks in elite football, correlating these measurements with goalkeeper strategies and enhancing prediction models for goalkeeper behavior.

2.4 Limitations of Previous Approaches

The previously discussed methodologies exhibit distinct requirements and operational settings, ranging from camera specifications to precision of analysis and associated metrics, requiring the exploration of new paradigms.

The methodologies reviewed exhibit significant limitations, including dependence on controlled environments and precise setups for optical systems, sensor drift and calibration challenges in inertial systems, magnetic interference in magnetic systems, and dynamic camera uncertainties in image capture. Movement estimation struggles with occlusion and variability in poses, while accuracy and bias issues arise from marker placement and single researcher involvement. Commercial systems face constraints related to cost, accessibility, and customization, and aquatic sports analysis is hindered by data scarcity, camera motion complexity, and occlusion by bubbles. Addressing these limitations requires novel techniques and improved methodologies to advance the field of sports motion capture and movement analysis.

2.5 Unsolved Issues and Challenges

The scarcity of publicly accessible datasets capturing athlete swimming poses a significant obstacle to advancing `markerless` swimming analysis. The reluctance of many research studies to share their databases slows the progression, limiting the implementation of recent advancements such as the utilization of CNNs.

Furthermore, the challenge of working with two distinct cameras — one aerial and one underwater — poses a unique hurdle. It may need the fusion of information from both media to derive

performance metrics. Additionally, managing the movement of the cameras presents another challenge as it impacts the stability and consistency of image capture. The motion of the cameras, whether due to water currents or aerial adjustments, can induce blur, distortion, or variations in the captured frames.

Moreover, the motion of the swimmer generates bubbles that occlude the human body, posing challenges for pose estimation in certain frames. Finally, the oscillation of skin masses can lead to imprecise joint detection.

2.6 Summary

This chapter presents a comprehensive overview of current advancements and challenges in motion capture technologies applied to sports evaluation. It covers optical and non-optical motion capture systems, detailing marker-based and `markerless` techniques, along with their respective applications and limitations. Optical systems, such as passive and active marker-based setups like Vicon and CODA, offer robust data collection capabilities but require controlled environments and post-processing for marker dropout. `Markerless` systems, exemplified by `Theia3D`, provide flexibility in data collection without markers, facilitating naturalistic sports analysis albeit with challenges in occlusion handling. Non-optical methods like inertial and magnetic systems offer portability and cost-effectiveness but face issues like sensor drift and underwater distortion. The chapter underscores the need for innovative solutions to enhance accuracy and applicability in diverse sports settings, highlighting ongoing challenges like occlusion management and dataset availability in aquatic sports analysis.

Chapter 3

A Toolbox for Swimming Analysis

This research works with underwater videos captured by a commercial, and off-the-shelf camera apparatus where the subject of interest, the swimmer, is not fitted with specific body markers, and is therefore non-intrusive. The videos used in this study were provided by the University of Porto's Biomechanics Laboratory (LABIOMEP-UP).

We now describe the structure and technologies used in this image-processing work-flow used to automatically extract performance metrics of interest.

3.1 Toolbox Work-Flow Outline and Image Processing Libraries

As depicted in the figure below 3.1, our image-processing architecture follows a simple and classic work-flow comprising of two major analysis phases.

The input consists of a raw, un-edited AVI-formated digital video, typically lasting between 60 and 120 seconds with frame rate of 119 *frames-per-second (fps)* with a frame of 1090-by-1080 pixels images.

The first, the *pre-processing* phase, removes a prologue and epilogue video sections where the subject of interest is inactive for the purpose of the analysis. These segments can last between 5 and 10 seconds and corresponds to the periods where the swimmer is getting ready to perform. This phase also defines a key reference line - *waterline* - crucial for calculating metrics, and *regions of interest*, which refine key-points to focus solely on the swimmer being evaluated.

The final phase, *metrics extraction*, focuses on deriving key performance indicators using the filtered key-points and image references obtained in the preceding phase. These metrics include stroke frequency, arm depth during strokes, and leg-kick frequency.

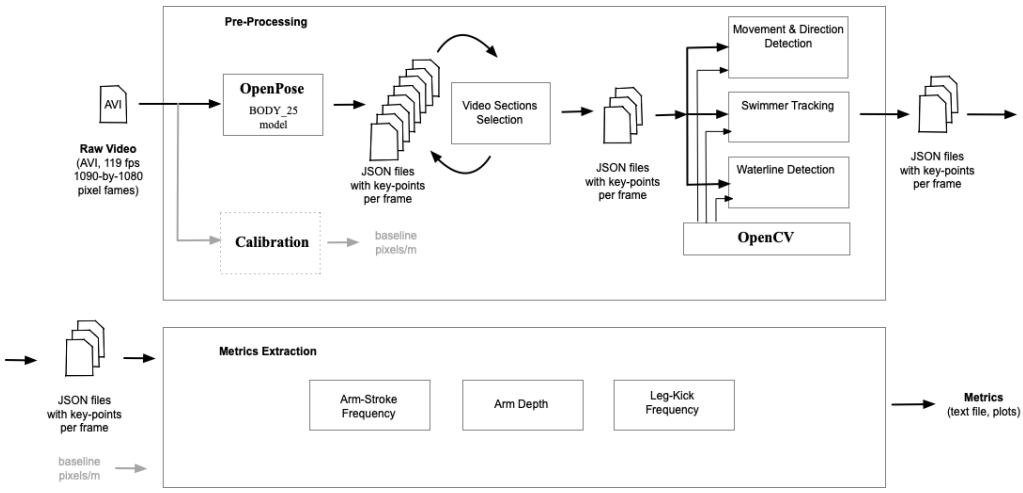


Figure 3.1: Markerless Swimming Analysis Work-Flow.

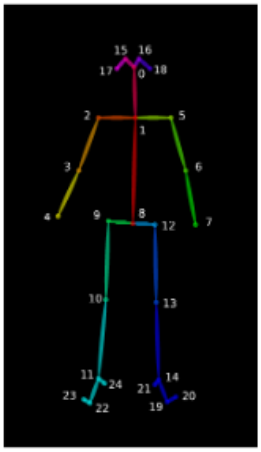
To support all these phases, the *pre-processing* starts with a comprehensive video analysis, to detect the presence/absence of activity and uncover, in a preliminary fashion, basic swimmer activity. This analysis, using `OpenPose`, extracts key-points of human activity in the water that support a variety of other analysis in subsequent phases as described below.

We now briefly outline the `OpenPose` and `OpenCV` libraries, highlighting the various approaches used to leverage the information extracted during the subsequent workflow phases.

3.1.1 `OpenPose`: a Toolkit for Human Pose Extraction

This toolkit includes a library that estimates the poses of human subjects in a given image based on a given input modelAvailable on CMU-Perceptual-Computing-Lab on input videos of swimmers. The chosen human model is referred to as `BODY_25`, illustrated in Figure 3.2, as it is noted in the literature to be more accurate than the COCO and MPI models[1]. The `BODY_25` model represents 25 key-points of the human body, numbered as shown in the Figure 3.2 with the corresponding `JSON` key-points definition file.

[1] While the COCO is a 18-point model and hence less accurate than the BODY_25 model, the 15-point MPI model is not supported in the current release of `OpenPose`.




```
{
    "version": X,
    "people": [
        {
            "pose_keypoints_2d": [],
            "face_keypoints_2d": [],
            "hand_left_keypoints_2d": [],
            "hand_right_keypoints_2d": [],
            "pose_keypoints_3d": [],
            "face_keypoints_3d": [],
            "hand_left_keypoints_3d": [],
            "hand_right_keypoints_3d": []
        }
    ]
}
```

Figure 3.2: `BODY_25` `OpenPose` Human Model (image taken from (17)) and corresponding `JSON` definition file.

The output of a pose estimation is a set of `JSON` files, one for each video frame with the co-ordinates of selected key-points, such as the face, left- and right-hands or knees. The coordinates, referred to as pixel (x,y) frame coordinates are associated with a real value (between 0.0 and 1.0) reflecting the level of detection confidence of the corresponding anatomic key-point of the model.

3.1.2 `OpenCV`: a Library for Computational Vision

The `OpenCV` (Open Source Computer Vision Library) (20) is a library of programming functions mainly for real-time computer vision. These computer vision tasks include methods for acquiring, processing, analyzing and understanding digital images, and extraction of high-dimensional data from the real world in order to produce numerical or symbolic information, *e.g.*, in the forms of decisions. It is written in `C++` but includes bindings for `Python` and `Java` and include thousands of image processing algorithms and more recently statistical ML models.

3.2 Pre-Processing

The key function of this work-flow phase is to clearly identify and ignore the video sections that correspond to the segments of the performance of the swimmer where he/she is preparing for his/her *trial*. Additionally, other challenging automation aspects are also addressed in this phase, as are the isolation of the *swimmer of interest* (as there can be more than one swimmer in the frame), detection of *movement of interest* and *waterline detection*, as the later is key for some of the performance metrics analysis.

The input is a full video file and the output is two-fold. First, the various `JSON` for each frame with the corresponding key-points, and second, using the information about the presence of the key-points, a set of videos corresponding to the sections of the original video with swimming activity.

3.2.1 Calibration

Although not developed in the context of this work, the workflow includes a place-holder for the important step of calibration of the image plane. This common calibration phase, makes use of a underwater checkerboard to allow to determine the ration of pixels per meter subsequently used in some of the key performance metrics, such as swimmer velocity. an illustrative example of an underwater checkerboard used for image calibration is depicted in Figure 3.3 below.

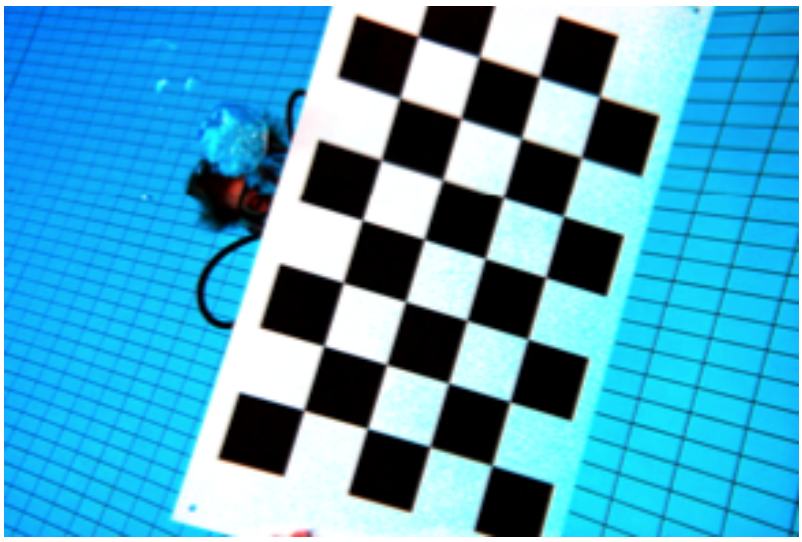


Figure 3.3: Illustrative example of underwater checkboard used for image calibration.

There has been extensive work on this particular topic (*e.g.* (1)) using underwater and surface cameras taking into account refraction. Time constrains prevented us to explicitly exploring this aspect of the workflow.

3.2.2 Video Sections Identification

We now describe the algorithm used to identify and isolate video segments with an active swimmer. To identify video segments where there is either no swimmer or the swimmer is stationary, we rely on frames where key-points are absent or in very low numbers.

To automate this process, we developed a script that repeatedly invokes the `OpenPose` binary using the selected human model to retrieve key-points of any human present in a frame. If a high number of consecutive frames fail to detect any key-points, an error message is generated, and the program stops without completing the full video analysis. The script repeatedly invokes this binary, using the failed frame as the starting point for the next video segment to be processed. This continues until no more *empty* frames are detected, thereby delimiting the sections of meaningful video. This approach thus results in multiple video segments, or a single video segment, from which the longest one was chosen for further processing, as it corresponds to the main section of the video with a swimmer.

3.2.3 Movement Detection

This step focuses on tracking the motion of the swimmer relative to the pool.

While we assume that the digital camera synchronously follows the swimmer [2], and that the speed of the swimmer is approximately equal to the the speed of the camera, we still need to detect when the camera is effectively moving relative to the pool.

To carry out this detection, we developed a movement detection algorithm that determines when the swimmer is moving in a steady fashion through the pool. To this effect, we relied on large pool weights, or *blobs*[3] that are located along the lanes of the pool. Lateral movement of the swimmer and thus of the camera, can therefore be detected when these *blobs* are observed moving across frames in distinct pixel positions.

The clear identification of these *blobs* underwater was, however, far from trivial which underscores the difficulty of any sub-aquatic analysis. To make the *blob* identification reliable we use *blob* detector function of the `OpenCV` by finding (through a non-trivial parameter-value exploration) of four of the parameters of this algorithm namely:

- `minCircularity`: Specifies the minimum roundness of the detected blobs.
- `minConvexity`: Ensures the detected blobs are convex.
- `minArea`: Sets the minimum area of the blobs to be detected.
- `minInertia`: Controls the minimum inertia ratio of the blobs.

After exploring these parameters values, we were unable to achieve the desired results due to the large number of false detections. To mitigate this issue, we applied a red filter to the video to detect only the *blobs*. However, the underwater position of the *blobs* caused a bluish tint, complicating the detection process. We attempted to enhance the red, but the issue persisted. To solve this we divided the video frames into four regions, with special interest on the bottom part (Figure 3.4).

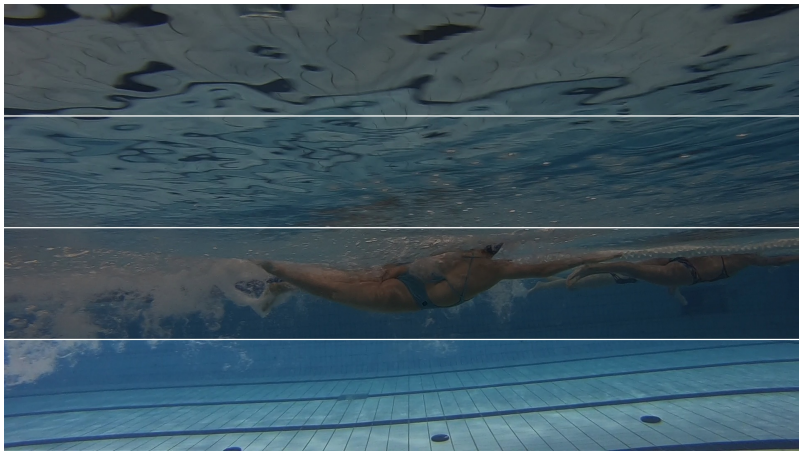


Figure 3.4: Frame divided horizontally in 4 analysis regions.

[2] It is unknown any possible *jitter* between the swimmer and the camera, so we assume perfect synchronicity.

[3] In this context, *blobs* refer to distinctive regions or clusters within an image that represent significant objects or features of interest, such as the localized markers used to track swimmer movement in underwater video analysis.

This significantly reduced the number of detections to an average of 2 to 3 *blobs* per frame, which was satisfactory as there are typically 2 to 3 visible markers per frame and by visually analysing represented correctly the markers or *blobs* (Figure 3.5).

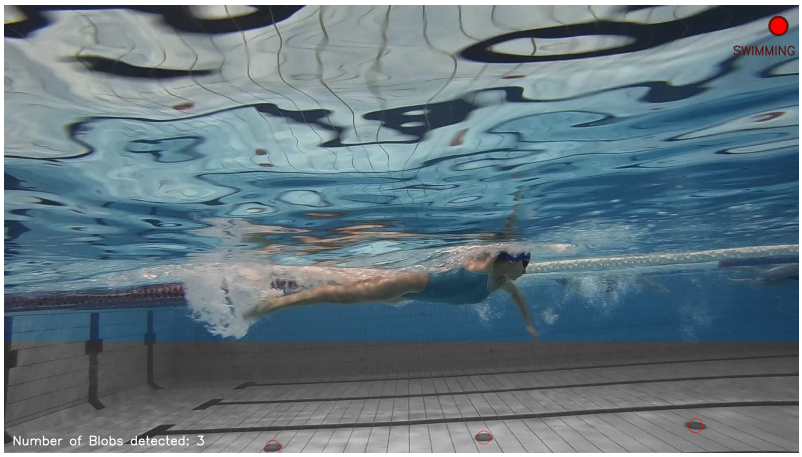


Figure 3.5: Example of the result of the *blob* detection algorithm.

To track the movements of the *blobs*, we relied on the `OpenCV` function to return the frame $x-y$ positions of the *blobs* which were then used to calculate the Euclidean distance between them in consecutive frames. As such we track the movement of objects across frames and the relative speed of the camera. Specifically, the horizontal distance between $x-y$ in consecutive frames is calculated to verify if there is movement in the opposite direction of the intended path of the swimmer.

An example of this movement is illustrated in Figure 3.6. Here, there is a horizontal displacement of the *blob* between two consecutive frames. The color green represents the current *blob* detection, while blue indicates the previous detection. This indicates that the same object had a displacement on the x-axis, reflecting the movement of the camera. However, due to some frames lacking or containing incorrect detections, we implemented additional criteria to ensure the resulting video had at least 5 seconds of continuous footage. This adjustment resulted in trimming the video only once, thereby enhancing the accuracy of the movement detection.

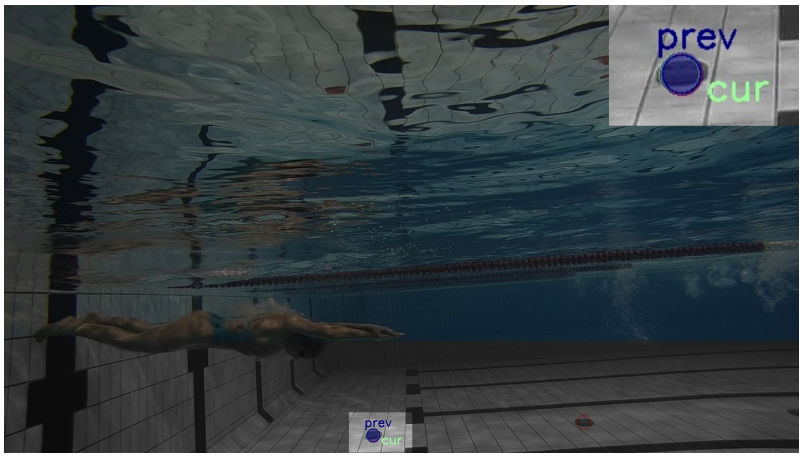


Figure 3.6: *Blob* detection across consecutive frames.

3.2.4 Movement Direction

In addition to movement, it is also important to determine direction (left to right or right to left). To this extent we analyze the first frame with key-points to determine their position relative to the sides of the frames. Key-points on the left suggest a left-to-right swimming motion, and vice versa, assuming the swimmer starts freestyle from a wall-leaning position. These methods collectively enhance the precision and reliability of automated swimming analysis (Figure 3.7).

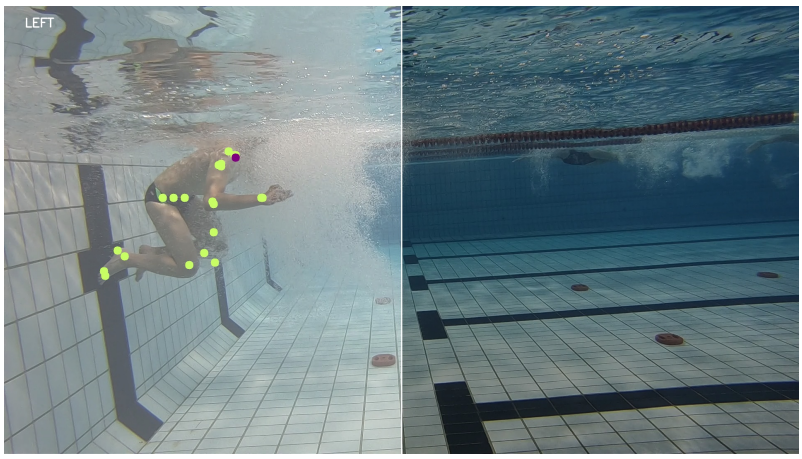


Figure 3.7: Key-points on the left of the screen suggesting movement from the left-to-right.

3.2.5 Identification and Swimmer Tracking

In some situations, videos can feature multiple swimmers, presenting a challenge in accurately tracking the main swimmer for metric evaluation (Figure 3.8). Another issue was incorrectly detecting reflections of people on the water (Figure 3.9). To address this, we adopted a method to isolate the main swimmer by selecting the key-point closest to the center of each frame. This strategy assumes the camera primarily focuses on following the main swimmer, allowing us to prioritize tracking their movements and metrics despite the presence of other swimmers.

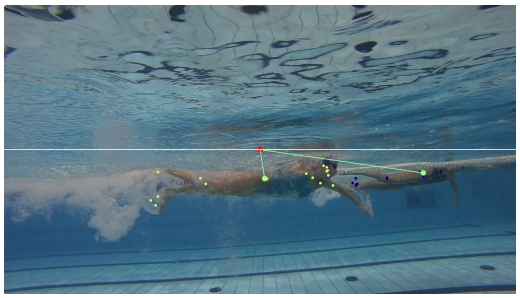


Figure 3.8: Detection of key-points of multiple swimmers.

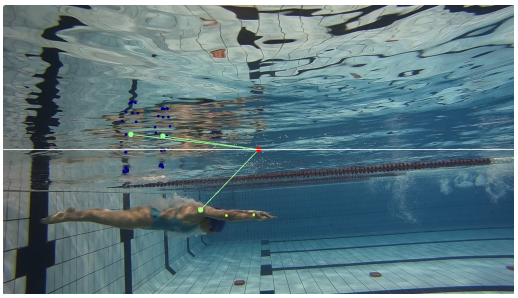


Figure 3.9: Detection of key-points of swimmer in the water reflection.

3.2.6 Waterline Detection

In the analysis of swimmer movements, is key to understand the relative position of its key-points relative to the waterline. This is the case of the *stroke* analysis where one needs to determine the exact moment when the hand of the swimmer touches the water after the aerial phase of the upper limb. Figure 3.10 below depicts a frame corresponding to the beginning of a stroke.

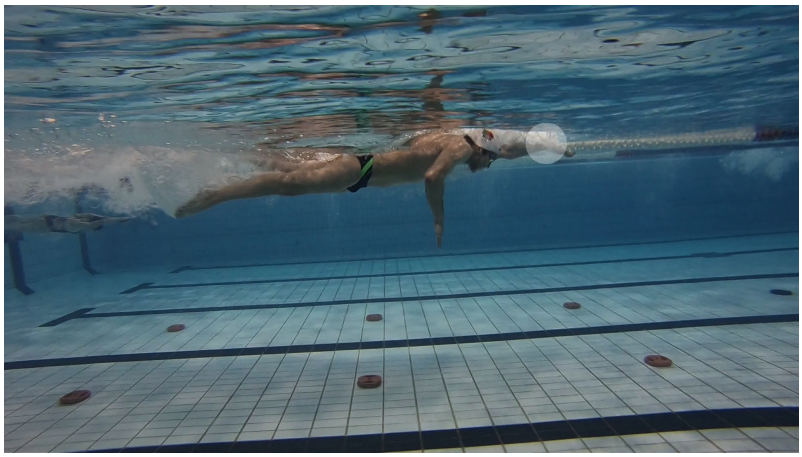


Figure 3.10: Beginning phase of an arm stroke with the hand key-point first coming into scene after an aerial phase.

To detect the waterline accurately we developed an algorithm that first detects all lines, using the Canny Edge detection (15) and Hough line detection (16) functions from `OpenCV`. After this step, it filters the lines to only retain horizontal lines. Although simple, this approach is still challenging, as there are numerous horizontal and vertical lines detected due to water surface refraction and the tiled composition of the swimming pool as depicted in Figures 3.11 and 3.12.

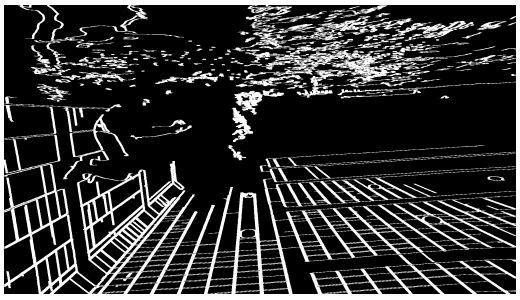


Figure 3.11: Edges detected using `OpenCV`'s Canny Edge function.

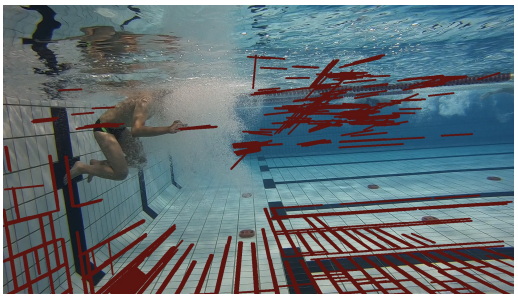


Figure 3.12: Lines detected using `OpenCV`'s HoughLinesP function.

The lane dividers in swimming pools — typically red and white — provide a reliable reference for approximating the waterline. By applying an additional filter to isolate red objects, specifically targeting the lane dividers, the algorithm effectively narrows down the candidates for the waterline (Figure 3.13).

Figure 3.13: Filter to highlight the swimming pool lane lines.

To further refine the waterline detection, the average y-coordinate of the hip key-points are used to localize the region where the waterline is likely to be, given that the hips of the swimmer are close to the surface of the water during swimming. This information, combined with a threshold set at one-third of the frame height, limits the search area for the waterline.

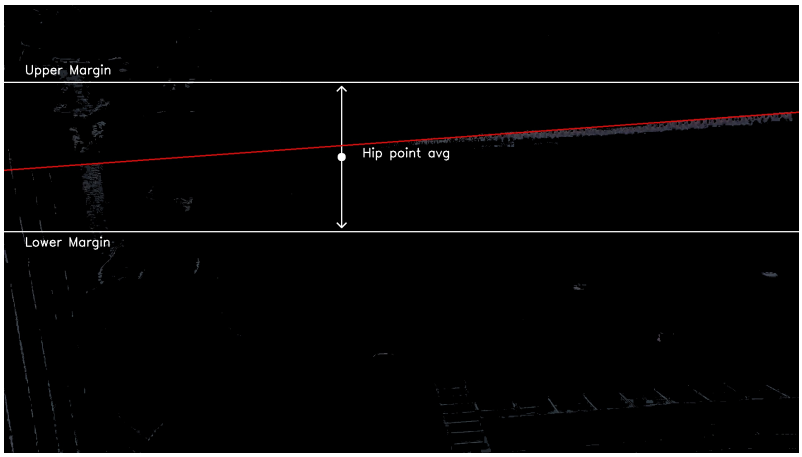


Figure 3.14: Illustration showing the calculated waterline with bounds and average hip key-point.

We used the `cv2.HoughLinesP` function from `OpenCV` to detect lines. If multiple lines were retrieved, we selected the line with the highest vertical position (*i.e.*, the line closest to the top of the image). This line is likely to be above the lane line and therefore represent the waterline.

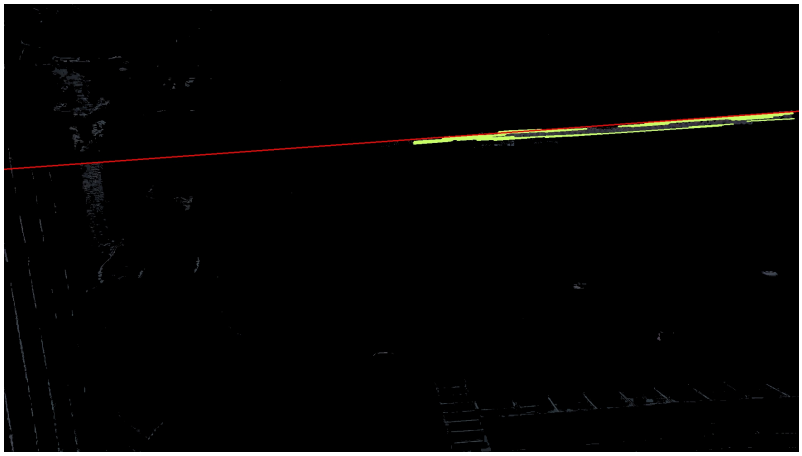


Figure 3.15: Lines detection result. The red line represents the highest line detected.

3.3 Extraction of Metrics

The final phase of the work-flow focuses on extracting key swimming performance metrics, which may vary depending on the tracking approach used: stroke frequency, kick frequency and depth of movement.

3.3.1 Stroke Frequency

The bounding of the waterline, highlighted in green in the Figure 3.16, allows us to safely detect a arm stroke when the key-point associated with the wrist key-points pass through this area. A naive algorithm would yield various consecutive detections of the beginning of the same arm stroke as

there are many consecutive frames where the wrist key-point are within the waterline bounding lines. To avoid this issue, once a first detection is observed, the algorithm ignores the following N frames to allow for the wrist to effectively travel below the waterline. In our experience, we used a percentage of the frames per second (*e.g.*, 40%) which proved to be an effective method to avoid the multiple detections of the same stroke.

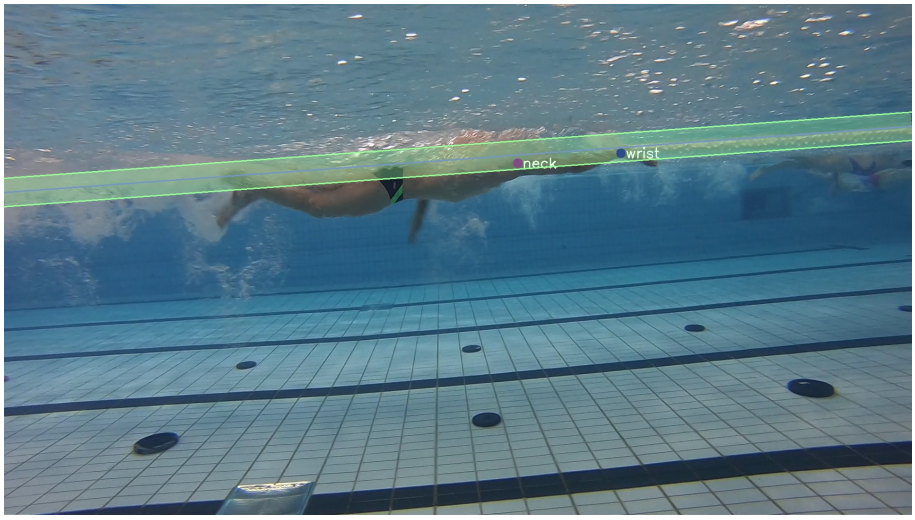


Figure 3.16: Example of the beginning of a stroke.

With such a simple algorithm, the point where the arm exits the water could also be incorrectly detected as an the beginning of an arm stroke. To address this issue, we imposed an additional constraint based on the direction of the swimmer. When swimming from left to right, wrist key-points appearing prior to the head are disregarded, and conversely, when swimming from right to left, wrist key-points appearing after the head are ignored. This refinement ensures a very accurate detection of stroke cycles.

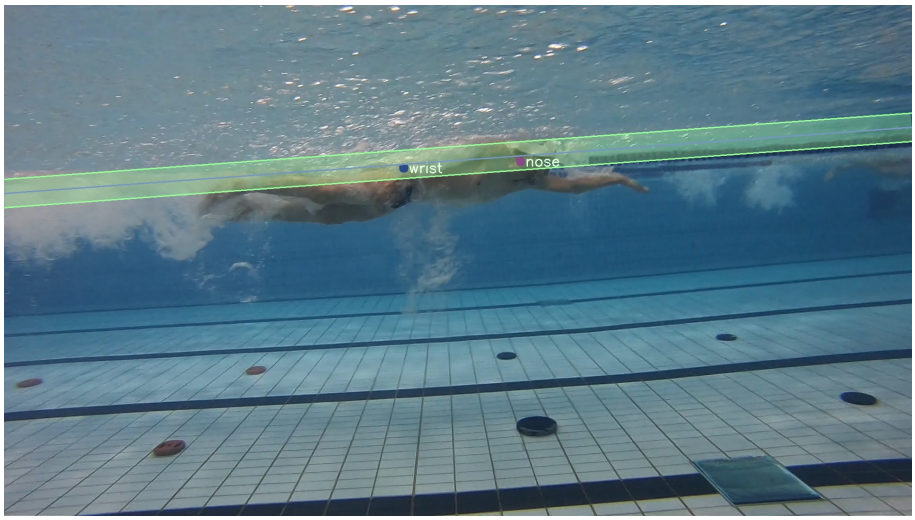


Figure 3.17: Example of the end of a stroke.

3.3.2 Stroke Arm Depth

The arm depth in a stroke refers to how deep the arm goes while performing a freestyle stroke, as illustrated in Figure 3.18. To calculate the arm depth, we averaged the depth of each cycle, determined by the arm stroke calculated previously. Each depth is defined as the distance between the wrist key-point and the waterline.

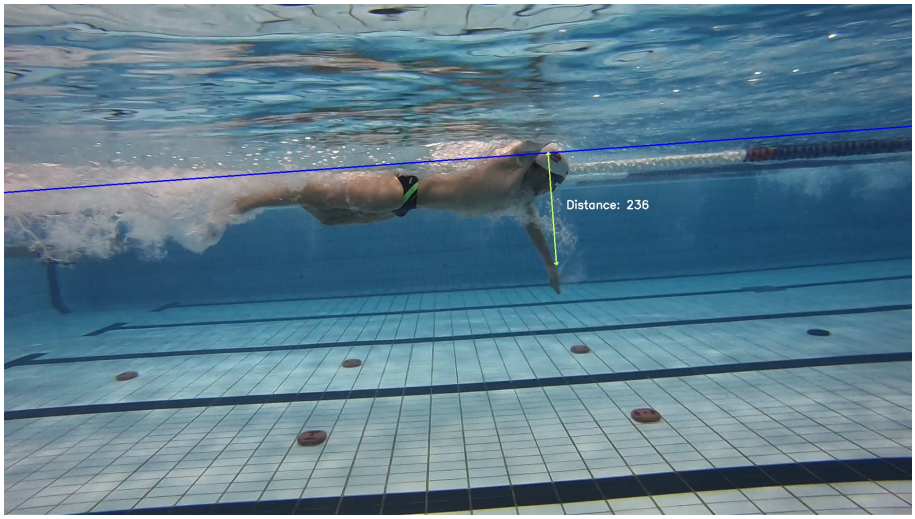


Figure 3.18: Arm depth illustration.

It is beyond this research work the determination of the depth in terms of metric value as this would require an accurate calibration of the pixel-to-distance. In this work we will only determine maximum arm extension in terms of number of image pixels.

3.3.3 Leg-Kick

The term *leg-kick* refers to the rhythmic movement of the legs used to propel the swimmer through the water. In freestyle this movement is called the flutter kick which involves a continuous and alternating up-and-down movement of the legs, typically performed from the hips with pointed toes. The kick is relatively small and rapid, generating propulsion primarily from the movement of the feet and lower legs.

To calculate the frequency of the leg-kick, the primary method involves analyzing the key-points of the ankles to detect changes in direction from up to down or down to up. The direction of the ankle (foot) movement is determined by comparing the y-coordinate of the current key point with the previous one. A lower y-coordinate indicates the foot was descending, whereas a higher y-coordinate indicates it was ascending.

A change in direction is identified by comparing the current direction with the next direction. The code primarily counts changes in direction from down to up because the ankle is more visible during this movement, thus have more key-points detected to be used in the algorithm. When the swimmer moves from left to right, the algorithm utilizes key-points corresponding to the left ankle and vice versa, leveraging their increased visibility (Figure 3.19).

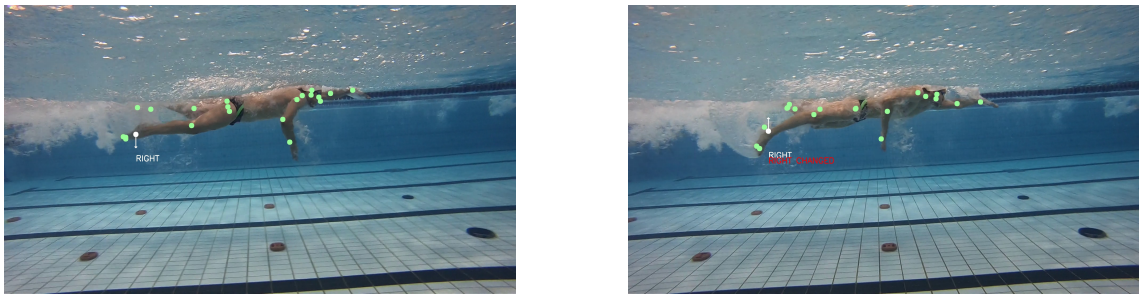


Figure 3.19: Illustration of foot direction change during leg kick (left: Foot moving downwards in the previous frame, right: Frame indicating the change of direction).

False positives occur when the swimmer is not actively swimming but triggers a leg-kick by moving their feet. This happens because the program monitors only the movement along the y-axis. This occurs frequently, when the swimmer is waiting to initiate its swim as depicted in Figure 3.20.

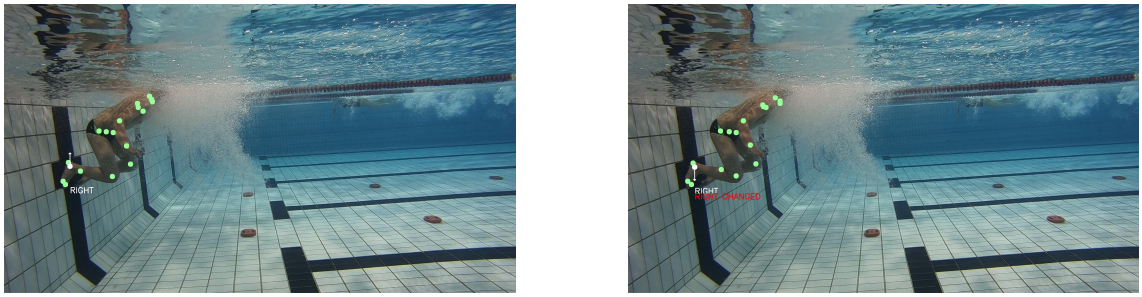


Figure 3.20: Inaccurate detection triggered while the swimmer was waiting to begin (two consecutive frames).

To eliminate these erroneous leg-kick detection, we refined our algorithm to filter change-of-direction frames to be within the interval between the first arm stroke and the last.

Multiple changes in direction detected during a single leg-kick result occasionally induce `OpenPose` to mistake the left for the right limb, as illustrated in Figure 3.21 using two consecutive frames. To try to mitigate this issue and enhance the accuracy of the leg-kick algorithm, linear interpolation was implemented on the missing key-points associated with the ankle key-points between two consecutive frames. However, this approach was discarded as it did not significantly improve accuracy, as discussed in Chapter 4.

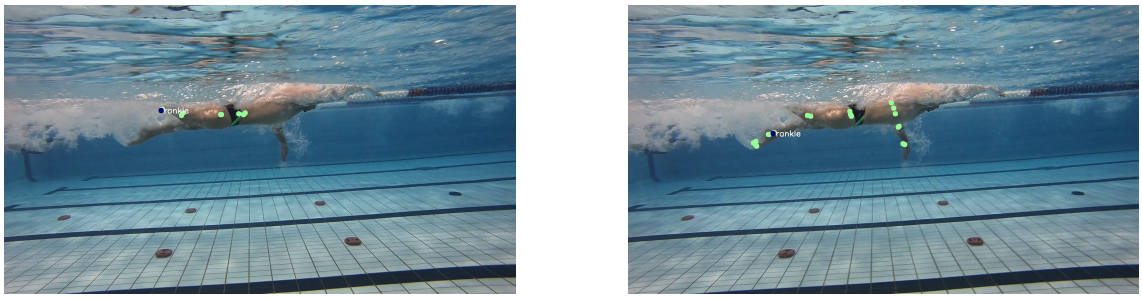


Figure 3.21: Illustration of left-right confusion in ankle identification during consecutive frames of a leg-kick. (left frame with left ankle mistakenly identified as the right ankle; right frame vice versa.)

A more effective approach included the addition of a time constraint between leg-kick detections and to consider the distance covered between kicks. This helps avoid detecting minor y displacements that do not represent actual kicks, thus enhancing the accuracy of the detection algorithm.

This method ensures that only significant leg movements within the defined time interval are counted as kicks, providing a more reliable measurement of leg-kick frequency. As a result, the precision improved from 37% to 70% in the video ID01, and from 41% to 67% in the video ID03.

3.4 Challenges

There were several technical challenges related to the installation and operation of the developed work-flow. First, and foremost, the biggest challenge was installing `OpenPose` on a local machine with a MacOSX operating system (v14.0 (23A344)) and without a Graphics Processing Unit (GPU) given that such high-performance computing infrastructure is crucial for the accurately extracting `OpenPose`'s model in useful time. Despite following multiple official and unofficial tutorials, we were unable to install it due to hardware constraints and the fact that Caffe, a necessary dependency, is disabled for the OSX's programming environment. The next attempt involved using free version of Google Colab with CPU-only support, but it took 30 minutes to process 1 second of video, making meaningful progress nearly impossible. We then tried the paid version, which provided GPU access, but monthly resource limitations were quickly exhausted, further hindering project advancement. Lastly, we explored various cloud providers with GPU resources, but many had significant restrictions. Each required support requests to create a virtual machine, and only Google Cloud Computing allowed us to set up a machine. However, even Google Cloud had regional resource limitations. After several attempts, we successfully created an NVIDIA T4 machine with CUDA, enabling us to run `OpenPose` effectively.

Concerning the development phase, the fact that we were dealing with underwater footage presented additional challenges. The unique lighting conditions, presence of bubbles and different setting and type of movement (swimming) of which the `OpenPose` model is trained to detect people affected the quality and consistency of the output (key-points).

3.5 Summary

We described here the structure and the implementation of the image processing work-flow that ultimately allows us to automatically derive key swimmer performance metrics. This analysis flow has been completely automated and requires minimal user intervention taking as input and un-annotated and unprocessed raw video. The next chapter presents empirical results of its use in real swimmer videos.

Chapter 4

Experimental Results

This chapter presents experimental results of the use of the developed work-flow and associated analysis algorithms for two (2) real freestyle *crawl* swimming training videos. Besides the key performance metrics of interest, we also present results that reflect the ability of `OpenPose` and the developed algorithms in detecting specific anatomic key-points over the course of the videos. We then use the detected key-point to evaluate the ability of the developed algorithm in automatically extracting key swimmer performance metrics such as stroke detection frequency, stroke duration, and the average depth of arm movements during its movement, leg kick frequency and number of kicks per arm stroke. Lastly, we compare the automated results with manual observations. This comparison assesses the accuracy of the algorithm in identifying arm stroke, leg movements and arm depth.

4.1 Key-Point Detection Ability

We begin evaluating basic detection statistics such as the number of frames where `OpenPose` failed to detect any key-points (*empty frames*), the average and maximum consecutive frames without points detected. The results for the selected two use case video are presented in Table 4.1.

Table 4.1: Empty Frames Statistics for the two input videos.

Video	**ID01**	**ID03**
Total Frames	1895	1846
Empty Frames	303 (16%)	207 (11%)
Average Consecutive Empty Frames	2.24 ± 1.88	1.95 ± 1.85
Longest Empty Frame Sequence	12	13

The number of *empty* frames is rather small, and so are the average number of consecutive *empty* frames and longest sequence of *empty* frames. Given that the frame rate of the video is 119 fps, 12 and 13 consecutive frames represent approximately 0.1 seconds, thus ensuring minimal impact on the overall analysis. It is particularly important that these parameters are low to minimize the interpolation error of missing key point data and consequent detrimental impact on overall accuracy of key point estimates.

Table 4.2 lists the average frame detection rate (in percentage) per key-point for the ID01 and ID03 uses-case videos .

Table 4.2: Frames percentage Detected per Key-point for ID01 and ID03 use-case videos.

	Video	
Key-point	**ID01**	**ID03**
Nose	55.98	51.63
Neck	74.34	73.97
Right Shoulder	65.82	68.56
Right Elbow	57.09	62.16
Right Wrist	48.89	53.06
Left Shoulder	64.02	61.94
Left Elbow	52.49	49.15
Left Wrist	47.14	44.95
Mid Hip	79.31	79.87
Right Hip	78.15	75.79
Right Knee	77.04	73.69
Right Ankle	72.17	69.17
Left Hip	78.62	76.01
Left Knee	77.94	74.57
Left Ankle	71.43	67.90
Right Eye	53.39	48.59
Left Eye	35.77	30.78
Right Ear	55.71	49.97
Left Ear	17.94	11.75
Left BigToe	38.36	37.62
Left SmallToe	31.96	31.77
Left Heel	47.09	47.93
Right BigToe	51.75	57.09
Right SmallToe	39.26	49.31
Right Heel	51.06	56.54

These results in Table 4.2 reveal that key-points such as Mid Hip (**79.31%**), Left Hip (**78.62%**), Right Hip (**78.15%**), Left Knee (**77.94%**), and Right Knee (**77.04%**) exhibit the highest detection percentages, very close to 80%. This is hardly surprising as these key-points primarily include hips and knees, which are prominently visible during swimming strokes. Conversely, key-points like Left Ear (**17.94%**), Left Eye (**35.77%**), Left Small Toe (**31.96%**), Left BigToe (**38.36%**), Right Small Toe (**39.26%**) have lower detection rates, often due to their smaller size and frequent

occlusion by bubbles or limbs during swimming movements. The eyes and ears key-points are particularly obstructed by the arm of the swimmer during strokes.

Other less *significant* key-points, such as the *ear* have lower detection percentage, but also a great asymmetry between the detection values for the two sides, indirectly reflecting the direction of the movement of the swimmer. For example, the left ear and the right ear have remarkable different detection percentage values.

Table 4.3 presents the detection percentage of the various key-points comparing the left and right sides of the swimmer for the two videos. In these two videos the swimmer moves from left to right in the footage, so as can be expected, `OpenPose` exhibits higher detection percentage for the right-side variants for the relevant key-points.

In video ID01, this is evident in the detection percentages of key-points such as the Shoulder (**65.82%**), Elbow (**57.09%**), and Wrist (**48.89%**). Notably, the Knee (**77.94%** on the left vs. **77.04%** on the right) shows a negligible difference of about 0.9%, likely influenced by the camera angle. The disparities in detection percentages for the eyes and ears (**35.77%** vs. **53.39%** for left and right eyes and **17.94%** vs. **55.71%** for left and right ears) are also aligned with our expectations, reflecting when these key-points are visible to the camera. The wrists and elbows maintain around 50% detection, alternating visibility as the swimmer executes strokes, similar to the visibility of the heels during leg movements.

Table 4.3: Key-points Detection Percentages: Left-Side vs. Right-Side.

	Video: ID01		**Video: ID03**	
Keypoint	**L (%)**	**R (%)**	**L (%)**	**R (%)**
Shoulder	64.02	**65.82**	61.94	**68.56**
Elbow	52.49	**57.09**	49.15	**62.16**
Wrist	47.14	**48.89**	44.95	**53.06**
Knee	**77.94**	77.04	**74.57**	73.69
Ankle	71.43	**72.17**	67.90	**69.17**
Eye	35.77	**53.39**	30.78	**48.59**
Ear	17.94	**55.71**	11.75	**49.97**
Big Toe	38.36	**51.75**	37.62	**57.09**
Small Toe	31.96	**39.26**	31.77	**49.31**
Heel	47.09	**51.06**	47.93	**56.54**

For the ID03 video we observe a similar trend with Mid Hip (**79.87%**), Left Hip (**76.01%**), Right Hip (**75.79%**), Left Knee (**74.57%**), and Right Knee (**73.69%**) again showing the highest detection percentages. Conversely, Left Ear (**11.75%**), Left Eye (**30.78%**), and Left Small Toe (**31.77%**) exhibit notably lower detection rates, underscoring the challenges in consistently capturing smaller or occluded key-points.

Overall, these results, underscore the ability of *OpenPose* to detect key-point detection in most frames. To best illustrate this, we depict in figures below, the spatial distribution of all key-points

for all the detected frames, offering a comprehensive overview of the positions within the frame where the respective key-points were detected.

In this sample of images, we present only the plots related to the *wrist* and *ankle*, as these key-points are the most significant for detecting the movements of interest—namely, arm strokes and leg kicks.

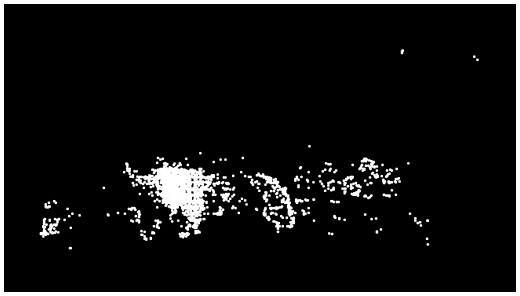


Figure 4.1: Spatial distribution of the detected ankles key-points (video ID01).

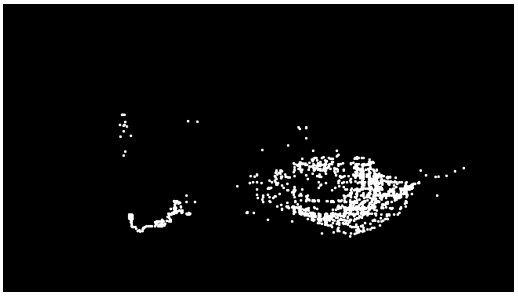


Figure 4.2: Spatial distribution of the detected wrists key-points (video ID01).

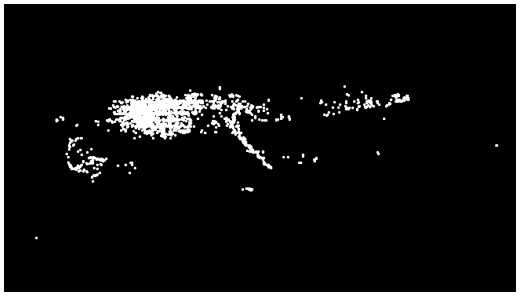


Figure 4.3: Spatial distribution of the detected ankles key-points (video ID03).

Figure 4.4: Spatial distribution of the detected wrists key-points (video ID03).

For the *wrist* plots (right-hand-side) (Figures 4.2 and 4.4) it is clear the circular movement of the hand of the swimmer, whereas for the *ankle* plots (left-hand-side) (Figures 4.1 and 4.3) where we can still observe a clear downward movement (the kick) alongside a more stable of key-points that clearly corresponds to the phase of the swim where the leg is extended and more stable.

Some points of the ankle are misidentified on the right side of the image, corresponding to instances when `OpenPose` internal representation switches the direction of the model of the body.

Analyzing the spatial distribution of all key-points, and assuming that the camera follows the swimmer, it becomes evident that `OpenPose` retrieves the key-points of the swimmer with a satisfactory accuracy to proceed with further analysis. This is indicated by the fact that the spatial distribution shows that swimmer occupies the center of the screen as expected.

Additionally, the horizontal orientation of the key-points and the traces of the arms and legs indicate a swimming motion, leading to the conclusion that the pose estimation is satisfactory. However, it is important to note the presence of some outliers, such as those seen limited in green in the following figures 4.5 and 4.6.

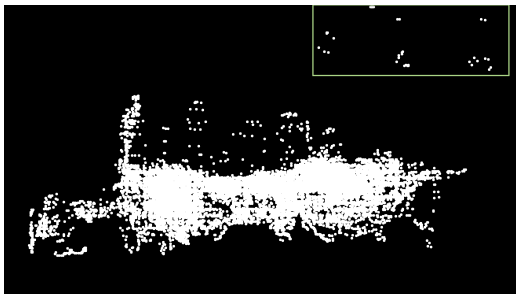


Figure 4.5: Spatial distribution of key-points with outliers highlighted related to the video ID01.

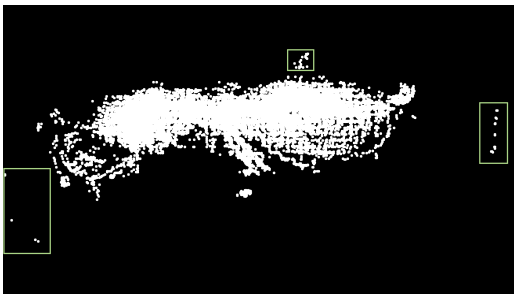


Figure 4.6: Spatial distribution of key-points with outliers highlighted related to the video ID03.

4.2 Stroke Frequency

We use the approach described in section 3.3 to automatically determine the swimmer's *stroke* frequency. In addition, we also extracted metrics using a manual frame-by-frame analysis of the video.[1] The results of the program are then evaluated by comparing these *stroke* frequency and other relevant data (such as number of frames in a *stroke*) to those retrieved by the automatic program. A difference within 50 frames between manual and automatic results were considered a match. As the videos rates (fps) is 119, 50 frames represent approximately 0.42 seconds, and which corresponds to a very tight detection criterion.

The evaluation metrics shown in Table 4.4 include the following:

- **Precision** as the ratio of correctly predicted positive observations over the total predicted positives. It is calculated as:

$$\text{Precision} = \frac{TP}{TP+FP}$$

  where TP is the number of true positives and FP is the number of false positives.

- **Recall** also known as sensitivity or true positive rate, is the ratio of correctly predicted positive observations to all the observations in the actual class. It is calculated as:

$$\text{Recall} = \frac{TP}{TP+FN}$$

  where FN is the number of false negatives.

- **F1-Score** is the harmonic mean of precision and recall, providing a balance between the two. It is calculated as:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

[1] To this extent we developed a simple program that allows the user to count *strokes* by pressing the space bar during a video. It stores the frames corresponding to the moments when the user pressed the space bar.

- **Accuracy** as the ratio of correctly predicted observations to the total observations. It is calculated as:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

where TN is the number of true negatives.

Table 4.4: Stroke Accuracy Results

Video	Precision	Recall	F1-score	Accuracy	TP	FP	FN
ID01	100%	90%	92%	95%	18	0	2
ID03	100%	82%	90%	82%	14	0	3

The stroke accuracy results for video ID01 indicate strong performance across all metrics. With perfect precision (100%) and a recall value of 90%, it achieves an F1-score of 92%. The high accuracy score of 95% reflects robust stroke detection capabilities, with only 2 false negatives identified, suggesting reliable identification of most strokes.

Similarly, for video ID03 a strong performance with perfect precision (100%) and a F1-score of 90% is also demonstrated. Despite a slightly lower recall of 82%, resulting in 3 false negatives, its overall accuracy stands at 82%. This underscores its capability to accurately detect strokes with reduced number of false positives.

To evaluate the arm *stroke* results we now examine the two videos in more detail. For the video ID01, we examined the source of the misdetection of two strokes, specifically strokes labelled as *stroke* 18 and 20. To assess why they were not detected, we trimmed the video to include only the frames corresponding to these *strokes*. We then plotted the detected key-points of both the left and right wrists throughout the video, along with the waterline and its margins, to determine if any key-points falls within that image band (Figure 4.7).

From the visualization of *stroke* 18, only two key-points were within the waterline margins. An analysis of their positions and the frames to which they belong, confirms that these key-points correspond to the end of the *stroke* which are not considered a false negative in the developed algorithm. For *stroke* 20, no key-points were within the waterline band region. The analysis of the video ID03 yielded similar results with 3 undetected *strokes*.

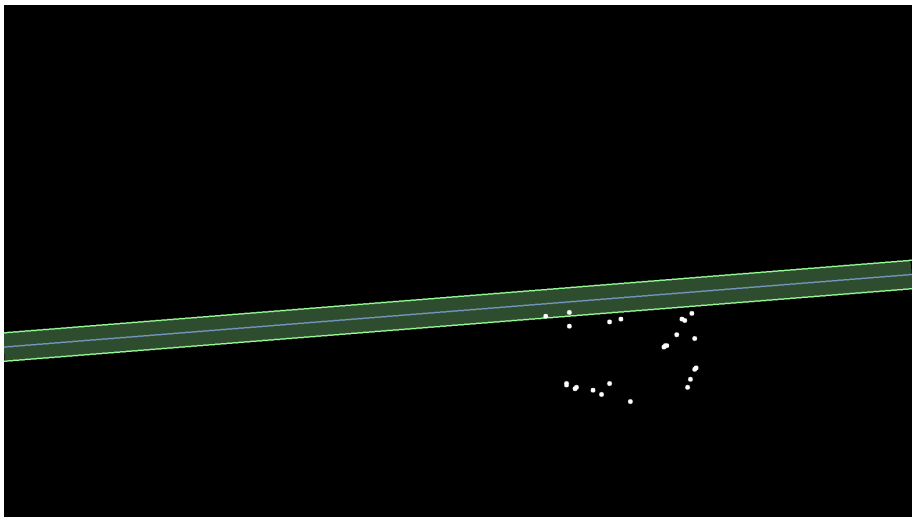


Figure 4.7: Illustration of wrist key-points during stroke number 18, along with the waterline and its respective bounds.

This analysis reveals that the reason for false negatives is due to the absence of wrist key-points, caused by `OpenPose`'s inability to detect the corresponding key-points on the region near the waterline. Clearly, this suggests that an improvement on the quality of the videos image frames could possibly increase the algorithm's accuracy.

Tables 4.5 and 4.6 that provide a detailed overview of the arm *strokes* detected in both videos. Each stroke is listed with its corresponding frame value, duration in frames, and duration in seconds, as well as a comparison between manually annotated (true) and detected values for each *stroke*. This table offers insights into the temporal characteristics of the detected arm *strokes*. The `Difference` column quantifies the deviation (in frames) between the manually annotated and detected values. This comparison highlights the performance of the algorithm in accurately identifying arm *strokes*, with occasional discrepancies noted between manual annotations and automated detection. Recall that 50 frames are less of half of a second of video time.

Of the detected *strokes* related to the video ID01, most exhibit relatively minor differences between the true and detected values, typically within 10 frames. However, *strokes* 16 and 17 show larger discrepancies of 42 and 12 frames, respectively. These larger differences may impact the accuracy of the stroke duration calculations.

Considering the threshold for discrepancy, *strokes* 16 and 17 are marginally within this limit, suggesting acceptable detection accuracy despite the discrepancies.

For the video ID03 we have similar results. Examining the detected *strokes*, most show relatively minor differences between the true and detected values, typically within 10 frames. However, strokes 2 and 11 show larger discrepancies of 21 and 12 frames, respectively. While these discrepancies are notable, they do not significantly impact the overall analysis.

These observations highlight the generally effective performance of the algorithm in identifying arm *strokes* within acceptable limits of discrepancy, underscoring its capability in stroke detection despite occasional larger deviations.

This performance is further corroborated by the overall arm *strokes* statistics in Table 4.7, which shows a high number of detected *strokes* per minute and reasonable consistency in *strokes* duration between manual and automatic measurements. In video ID01, the difference between manual and automatic measurements is residual. The difference is greater in video ID03 due to having more undetected frames. Each missed detection results in the previous *stroke* duration nearly doubling, thereby influencing the average *stroke* duration.

Clearly, a further refinement in detection algorithms could potentially reduce these deviations, ensuring more precise *stroke* identification. It is also important to note that undetected *strokes* represent opportunities for improvement in detection sensitivity, possibly influenced by factors such as occlusion or subtle motion variations.

Although based on just two videos, these results are very promising as they reveal an overall accuracy of the developed automated algorithm. The high precision, strong recall, and robust F1-scores demonstrate the reliability of the system in automated arm *stroke* analysis.

Table 4.5: Arm Stroke Analysis for Video ID01

Stroke Id	**True Value**	**Detected Value**	**Difference (Frames)**	**Duration (Frames)**	**Duration (Seconds)**
1	1351	1350	1	56	0.48
2	1413	1406	7	70	0.59
3	1486	1476	10	70	0.59
4	1550	1546	4	71	0.59
5	1625	1617	8	80	0.67
6	1691	1697	6	58	0.49
7	1769	1755	14	96	0.81
8	1829	1851	22	48	0.40
9	1902	1899	3	76	0.64
10	1968	1975	7	85	0.71
11	2047	2060	13	78	0.66
12	2125	2138	13	61	0.51
13	2202	2199	3	66	0.55
14	2272	2265	7	50	0.42
15	2341	2315	26	48	0.40
16	2405	2363	42	116	0.97
17	2491	2479	12	145	1.22
18	2558	-	-	-	-
19	2630	2624	6	-	-
20	2695	-	-	-	-

Table 4.6: Arm Stroke Analysis for Video ID03

Stroke Id	True Value	Detected Value	Difference (Frames)	Duration (Frames)	Duration (Seconds)
1	4092	4084	8	48	0.40
2	4153	4132	21	81	0.68
3	4221	4213	8	127	1.07
4	4285	-	-	-	-
5	4347	4340	7	64	0.54
6	4414	4404	10	69	0.58
7	4478	4473	5	81	0.68
8	4550	4554	4	49	0.41
9	4608	4603	5	64	0.54
10	4679	4667	12	69	0.58
11	4740	4736	4	133	1.12
12	4813	-	-	-	-
13	4875	4869	6	134	1.13
14	4946	-	-	-	-
15	5008	5003	5	69	0.58
16	5082	5072	10	71	0.60
17	5151	5143	8	-	-

Table 4.7: Stroke Statistics for videos ID01 and ID03.

Video	ID01	ID03
Strokes per Minute	100.9	94.39
Average Stroke Duration (Manual)	70.74 ± 6.89	66.19 ± 5.05
Average Stroke Duration (Automatic)	74.94 ± 25.29	81.46 ± 30.07

4.3 Arm Depth Movement

We use the approach described in section 3.3 to automatically determine the arm depth of the swimmer. In addition, we also manually extracted this metric by an exhaustive manual frame-by-frame analysis of the video.[2] The resulting averages of the manual approach are then compared against those obtained automatically.

Regarding video ID01, the analysis revealed small deviations, as shown in Figure 4.8, with an average manual measurement depth of 181 pixels and an automatic measurement depth of 192

[2]To this extent we developed a simple program that allows the user to use the mouse and manually determine the frame at which the arm is at its deepest position. This program then computes the minimum distance between the click and the waterline for each stroke cycle identified by the previous algorithm.

pixels, as detailed in Table 4.8. This difference of 11 pixels is considered insignificant, suggesting that both methods yield very comparable results.

Similarly, in video ID03, although the deviations are slightly higher, as seen in Figure 4.9, they remain minimal. The manual measurement of depth averaged 242 pixels, while the automatic measurement averaged 244 pixels, resulting in a difference of only 2 pixels, which is also deemed insignificant (Table 4.9).

Table 4.8: Arm Depth Values (Video ID01).

Arm Stroke	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**Avg**
Manual Depth	183	218	186	187	179	188	176	188	139	183	183	185	188	201	195	201	204	226	228	191.47 ± 19.82
Auto Depth	123	207	182	183	164	180	150	185	146	180	160	179	144	203	181	183	229	221	238	180.9 ± 29.81
Difference	60	11	4	4	15	8	26	3	7	3	23	6	44	2	14	18	25	5	10	10.6

Table 4.9: Arm Depth Values (Video ID03).

Arm Stroke	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**Avg**
Manual Depth	266	240	257	234	263	245	228	231	228	238	246	236	245	235	264	251	244.19 ± 12.77
Auto Depth	239	243	235	237	243	237	209	224	230	250	231	260	242	265	246	273	241.5 ± 15.66
Difference	27	3	22	3	20	8	19	7	2	12	15	24	3	30	18	22	2.7

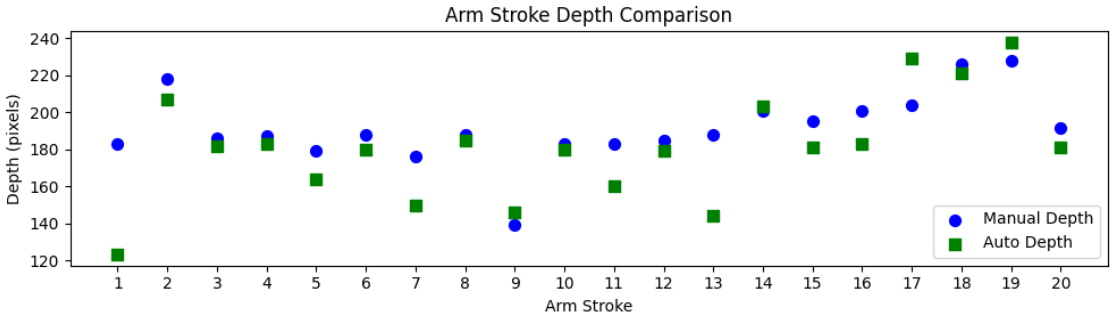


Figure 4.8: Arm Stroke Depth Comparison ID01

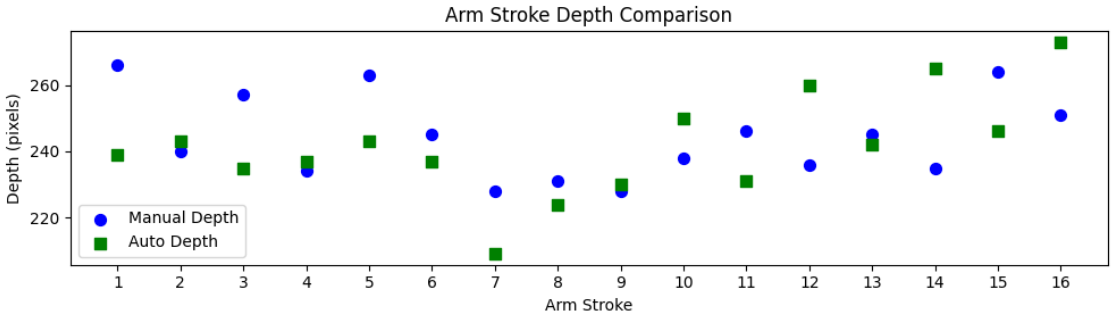


Figure 4.9: Arm Stroke Depth Comparison ID03

It is worth noting that the manual identification of the exact pixel where the wrist is located can also have inherent errors, which may contribute to the observed difference.

4.4 Leg Kick Frequency

We use the approach described in section 3.3 to automatically measure the leg kick frequency. We also calculated the frequency manually through a manual frame-by-frame analysis of the video[3]. Since the algorithm for counting the leg kicks only utilized the down-to-up movement, the same strategy was employed for the manual identification of leg kicks.

The results show different phases of the algorithm development for leg kick frequency identifications. To achieve the best results, we had some exploratory phase of the parameters used in the developed algorithm: duration of a leg kick and the minimum distance covered by a leg kick. The values used for the duration were between 10 and 40 frames with a step of 10 frames and for the minimum distance covered 0 up to 90 pixels with a step 10 pixels.

The results below represent the five best outcomes based on precision, showing an improvement that doubles the precision of the worst result, supported by the accuracy of the kicks per stroke. The results also represent the five best outcomes based on precision (Tables 4.10 and 4.11). As can be seen the precision of the worst result has improved twofold, as supported by the accuracy of the kicks per stroke. The best results correspond to the minimum distance covered by 40 pixels. It also shows the distance covered does not affect the results, so it can be discarded.

Table 4.10: Performance metrics ID01

Dur.	**Dist.**	**Leg Kick Freq.**	**P (%)**	**R (%)**	**F1 (%)**	**Accuracy (%)**	**Kicks per Stroke**
40	0	38	72	70	71	70	1.53
40	10	38	72	70	71	70	1.53
40	20	38	72	70	71	70	1.53
40	30	38	72	70	71	70	1.53
40	40	38	72	70	71	70	1.53
...	...	...	...	...	...	...	...
10	60	61	42	86	56	86	3.94

[3]To this extent, we developed a simple program that allows the user to count a leg kick by pressing the space bar during a video. It stores the frames corresponding to the moments when the user pressed the space bar.

Table 4.11: Performance metrics ID03

Dur.	Dist.	Leg Kick Freq.	P (%)	R (%)	F1 (%)	Accuracy (%)	Kicks per Stroke
40	0	36	67	57	62	57	1.5
40	10	36	67	57	62	57	1.5
40	20	36	67	57	62	57	1.5
40	30	36	67	57	62	57	1.5
40	40	36	67	57	62	57	1.5
...	...	...	...	...	...	...	...
10	10	80	36	93	52	93	4.58

Additionally, we evaluated whether interpolation of ankle key-points could enhance the accuracy of our calculations. Tables 4.12 and 4.13 compile the five best results. As can be seen, the precision is **57%** with interpolation compared to **72%** without interpolation. Similarly, the recall is lower at **57%** with interpolation compared to **70%** without interpolation. Regarding the results of video ID03, the precision decreases with interpolation, and the improvement in recall is not significant enough to consider it a valid approach.

Table 4.12: Performance metrics ID01 - Using interpolated key-points.

Dur.	Dist.	Leg Kick Freq.	P (%)	R (%)	F1 (%)	Accuracy (%)	Kicks per Stroke
40	0	43	57	57	57	57	1.58
40	10	43	57	57	57	57	1.58
40	20	43	57	57	57	57	1.58
40	30	43	57	57	57	57	1.58
40	40	43	57	57	57	57	1.58

Table 4.13: Performance metrics ID03 - Using interpolated key-points.

Dur.	Dist.	Leg Kick Freq.	P (%)	R (%)	F1 (%)	Accuracy (%)	Kicks per Stroke
30	0	38	64	64	64	64	1.75
30	10	38	64	64	64	64	1.75
30	20	38	64	64	64	64	1.75
30	30	38	64	64	64	64	1.75
30	40	38	64	64	64	64	1.75

Using the best precision results also shows high accuracy of the number of kicks per stroke, showing minimal differences from the manual detection, as seen in Table 4.14. The deviations are minor, corresponding to decimals, while using the worst precision could result in discrepancies of up to 3 kicks.

Table 4.14: Stroke Statistics for videos ID01 and ID03.

Video	ID01	ID03
Average Kicks per Stroke (Manual)	1.47 ± 0.51	1.44 ± 0.51
Average Kicks per Stroke (Automatic)	1.58 ± 0.51	1.5 ± 0.51

While our overall results are promising, distinguishing between the left and right legs poses challenges for `OpenPose`, especially in swimming contexts where the swimmer is in the sagittal plane and one leg occludes the other. Our algorithm faced particular challenges in this regard. Specifically, identifying the correct leg movements—such as distinguishing between upward and downward motions—can lead to inaccuracies when `OpenPose` misidentifies left and right legs. This mis-identification can result in multiple counts for a single leg kick.

4.5 Discussion

In this chapter, we analyzed the experimental results of applying the developed workflow to two swimming training videos. The results focused on the effectiveness of key-point detection, and comparisons between automated and manual observations for different performance metrics.

For ability to detect the key-points, the `OpenPose` using one of its models demonstrated high accuracy in detecting the anatomical elements of swimmers in most video frames. The detection was generally consistent, with occasional failures mainly due to rapid swimmer movements, temporary obstructions, or lighting variations. Visual representations confirmed the spatial accuracy and consistency of the detected key-points.

The automated stroke detection frequency closely matched the manual observations. This indicates that `OpenPose` is effective in identifying the number of strokes made by the swimmer. Minor discrepancies were observed, which could be addressed through further refinement of the `OpenPose` model. In order to improve the recall rate one could use interpolating techniques to ensure completeness of key-points sequence and hence enhance arm stroke detection accuracy. However, this approach faces challenges, particularly during the aerial phase where the absence of the key-points related to the arm complicates interpolation. When key-points are missing at the beginning of a stroke but data is available at the end and after the stroke, simple linear interpolation may create a straight line between the last and next key-point, which may not accurately reflect the actual motion.

The automated process to calculate the duration of each stroke, the average depth of arm movements and the leg kicks per arm stroke showed a strong correlation with manual measurements. This consistency suggests that the algorithm accurately captures the timing of arm movements, essential for analyzing swimming efficiency and technique.

The algorithm to calculate the frequency of leg kicks went through different exploratory variants to fine tune specific algorithmic parameters. The results obtained using the final version were compared against results obtained manually, showing a high degree of accuracy in identifying leg kicks. Simple linear interpolation in this context was discarded due to unsatisfactory results. The use of more advanced interpolation methods was left for future work.

In conclusion, `OpenPose` has proven to be an effective tool for extracting the described metrics, which were the primary focus of this research. This is despite the fact that little filtering or image processing in terms of enhancing the quality of the image was done. The various algorithms used, were effective in determining the stroke frequency, arm depth and leg kick frequency.

Chapter 5

Conclusion

The literature review highlighted various challenges and a lack of research and development in aquatic sports activity motion capture solutions, emphasizing the need for innovative approaches to overcome obstacles like camera positioning, occlusion, key metrics extraction. This research introduced a suite of digital image processing algorithms aimed at providing a practical and effective method for analyzing swimming performance focusing on low-cost solutions using common digital cameras and open-source software. The, although limited experimental results presented, underscores its potential for practical applications in swim performance assessment, suggesting it could be a valuable tool for coaches, athletes, and researchers seeking accurate and efficient analysis of swimming techniques.

5.1 Research Questions Revisited

In light of the experimental methodology and experimental results presented previously, we now revisit and comment the research questions outlined earlier in this document.

RQ1: Can the *OpenPose* models retrieve a significant number of anatomic key-points of interest of the swimmer's position that correlate with relevant performance metrics?

The findings indicate that `OpenPose` models can indeed retrieve a significant number of anatomic key-points. These key-points correlate well with relevant performance metrics, making the model a reliable tool for further analysis and validation. It is important to note that no manipulation of the videos was required during this process.

RQ2: In particular, is the use of a single commercially off-the-shelf camera (of reasonable image quality) a low-cost solution with acceptable cost-benefit?

Yes, as demonstrated in this work, good results can be achieved using a single commercially off-the-shelf camera. This approach provides a cost-effective solution while maintaining the necessary quality for effective analysis even without applying preprocessing and image enhancements algorithms.

RQ3: Is the approach too computationally expensive to be deemed practical?

While setting up a machine with a GPU can be costly and challenging, as described in section 3.4, it is necessary to process the videos within an acceptable time frame. However, once the initial processing with `OpenPose` is complete, the subsequent analysis and handling of results can be efficiently carried out on any standard machine without significant performance loss. Therefore, despite the initial setup cost, the overall approach remains practical and feasible for broader use, balancing cost and efficiency effectively.

RQ4: Is the accuracy of the derived performance metrics comparable to the ones derived by manual inspection of the video recordings of the sports activity?

The accuracy of the performance metrics derived from the automated analysis is comparable to those obtained through manual inspection. Although for a very limited sample of videos, this conclusion is supported by multiple manual validations, which suggests that the automated approach is very consistent with manually derived metrics.

Clearly, a more throughout investigation of the validation of the developed automated method needs to be pursued. One aspects should investigate the accuracy of multiple manual efforts of trained operators in identifying selected body parts in swimmer for videos with a comparable quality. We only had the benefit of a single manual operator and thus have considered out manual results are the golden reference.

5.2 Future Work

We now outline several potential directions for future work as presented in the next sections. The primary areas of focus include enhancing image quality, refining algorithms, expanding metric extraction, applying techniques to different swimming styles, utilizing neural networks, distinguishing body sides, and improving key-point correction.

5.2.1 Improving Image Quality for Enhanced `OpenPose` Results

A critical area for future work involves enhancing the quality of images used in the analysis. Improved image quality could potentially impact the accuracy of `OpenPose` in detecting and tracking key-points. Implementing advanced image pre-processing techniques, such as noise reduction, contrast enhancement, and color correction, could help achieve this.

5.2.2 Enhancing Metric Algorithms

To further refine the analysis, there is a need to improve the algorithms used for calculating various metrics. Enhancements could include:

- Developing more sophisticated algorithms that better account for the dynamic nature of swimming movements, utilizing advanced bio-mechanical knowledge.
- Integrating machine learning techniques to automatically adjust and optimize the algorithms based on a larger dataset of swimming activities.

5.2.3 Extracting Additional Metrics

Future research could focus on extracting a broader range of metrics to provide a more comprehensive analysis of swimming performance. Examples of additional metrics that could be extracted include:

- Aerial and subaquatic phase duration, providing additional insights into the efficiency of the swimmer.
- Body roll and pitch angles to understand the technique of the swimmer and body positioning.
- Joint angles and their coordination to identify potential areas for technical improvement.

5.2.4 Application to Different Swimming Styles

The techniques developed in this study could be adapted and applied to other swimming styles. This would involve:

- Extracting metrics from other swimming styles, such as butterfly, backstroke, and breaststroke.
- Conducting style-specific analyses to understand the distinctive characteristics and requirements of each swimming style, potentially even automatically identifying the swimming style.

5.2.5 Utilizing Neural Networks

Incorporating neural networks and other artificial intelligence techniques presents a promising direction for future work. Potential applications include:

- Developing deep learning models that can automatically recognize and classify swimming styles and techniques from video footage.
- Using neural networks to predict and correct key-point positions, improving the overall accuracy of the analysis.
- Leveraging AI for real-time feedback systems that provide swimmers and coaches with immediate insights and suggestions during practice sessions.

5.2.6 Distinguishing Right and Left Sides

Enhancing the algorithm we have developed here to increase its accuracy in distinguishing between the right and left sides of the body is crucial for a more detailed bio-mechanical analysis. This could involve:

- Designing and training models specifically to identify symmetrical movements and differentiate between sides.
- Incorporating side-specific data to enhance the precision of the analysis, particularly for movements where symmetry is a key performance indicator.
- Utilizing this distinction to provide more granular feedback on technique and performance.

5.2.7 Improving Key-point Correction

Improving the correction of key-points detected by `OpenPose` is essential for enhancing the reliability of the analysis. Future efforts could focus on:

- Development of advanced correction algorithms that use contextual information from adjacent frames to improve key-point accuracy.
- Implementing machine learning models trained on large datasets to predict and correct potential key-point errors.
- Enhancing the robustness of key-point detection in challenging conditions, in the presence of more complex backgrounds.

5.3 Final Remarks

In conclusion, these areas of future work provide a roadmap for further advancing the analysis of swimming performance. By addressing these challenges, researchers an practitioners can develop more accurate, comprehensive, and useful tools for improving swimming techniques and outcomes, enabling targeted interventions and training programs based on insights gained from the analysis of different swimming styles.

Overall, this work supports the thesis that it is possible to used the `OpenPose` framework to develop a work-flow for the automated analysis of swimming and extract accurate performance metrics that are competitive to manual video analysis approaches in a fraction of the time and effort.

References

[1] Thiago Telles Fábio A.S. Dias Guido Baroni Ricardo M.L. Barros Amanda P. Silvatti, Pietro Cerveri. Quantitative underwater 3d motion analysis using submerged video cameras: accuracy analysis and trajectory reconstruction. *Computer Methods in Biomechanics and Biomedical Engineering*, pages 1240–1248, 2013.

[2] Elena Ceseracciu, Zimi Sawacha, Silvia Fantozzi, Matteo Cortesi, Giorgio Gatta, Stefano Corazza, and Claudio Cobelli. Markerless analysis of front crawl swimming. *Journal of Biomechanics*, 44(12):2236–2242, 2011.

[3] Shenming Feng and Haifeng Hu. Learning joint structure for human pose estimation. *ACM Trans. Multimedia Comput. Commun. Appl.*, 16(3), 2020.

[4] Nicola Giulietti, Alessia Caputo, Paolo Chiariotti, and Paolo Castellini. Swimmernet: Underwater 2d swimmer pose estimation exploiting fully convolutional neural networks. *Sensors*, 23:2364, 2023.

[5] Chenyu Gu, Weicong Lin, Xinyi He, Lei Zhang, and Mingming Zhang. Imu-based motion capture system for rehabilitation applications: A systematic review. *Biomimetic Intelligence and Robotics*, 3(2):100097, 2023.

[6] Tianyu He and Qi Luo. A survey of motion capture technology and its application in sports. *Springer International Publishing*, pages 854–859, 2019.

[7] Ginés Hidalgo and Hanbyul Joo. Cmu-perceptual-computing-lab, 2017. Accessed: 2024-06-28.

[8] Bas Van Hooren, Noah Pecasse, Kenneth Meijer, and Johannes Maria Nicolaas Essers. The accuracy of markerless motion capture combined with computer vision techniques for measuring running kinematics. *Scandinavian Journal of Medicine & Science In Sports*, 33:966–978, 2023.

[9] Robert M. Kanko, Jereme B. Outerleys, Elise K. Laende, W. Scott Selbie, and Kevin J. Deluzio. Comparison of concurrent and asynchronous running kinematics and kinetics from marker-based motion capture and markerless motion capture under two clothing conditions. *bioRxiv*, 2023.

[10] Azam Khalili, Vahid Vahidpour, Amir Rastegarnia, Wael M. Bazzi, and Saeid Sanei. Energy-efficient diffusion kalman filtering for multiagent networks in iot. *IEEE Internet of Things Journal*, 9(8):6277–6287, 2022.

[11] Winnie W. T. Lam, Yuk Ming Tang, and Kenneth N. K. Fong. A systematic review of the applications of markerless motion capture (mmc) technology for clinical measurement in rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 20:57, 2023.

[12] Alexander Mathis, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, 21(9):1281–1289, 2018.

[13] Masato Nakai, Yoshihiko Tsunoda, Hisashi Hayashi, and Hideki Murakoshi. Prediction of basketball free throw shooting by openpose. In Kazuhiro Kojima, Maki Sakamoto, Koji Mineshima, and Ken Satoh, editors, *New Frontiers in Artificial Intelligence*, pages 435–446, Cham, 2019. Springer International Publishing.

[14] OpenCV. Opencv, . Accessed: 2024-06-28.

[15] OpenCV. Canny edge detection, 2023. Accessed: 2024-06-28.

[16] OpenCV. Hough line transform, 2023. Accessed: 2024-06-28.

[17] OpenPose. Pose Output Format (BODY_25), 2020. Accessed: 2024-06-28.

[18] David Pagnon, Mathieu Domalain, and Lionel Reveret. Pose2Sim: An open-source Python package for multiview markerless kinematics. *The Journal of Open Source Software*, 7(77):4362, 2022.

[19] Guilherme Pinheiro, Xing Jin, Varley Costa, and Martin Lames. Body pose estimation integrated with notational analysis: A new approach to analyze penalty kicks strategy in elite football. *Frontiers in Sports and Active Living*, 4:818556, 03 2022.

[20] Kari Pulli, Anatoly Baksheev, Kirill Kornyakov, and Victor Eruhimov. Realtime computer vision with opencv: Mobile computer-vision technology will soon become as ubiquitous as touch interfaces. *Queue*, 10(4):40–56, apr 2012.

[21] Gui Quanan and Xia Yunjian. Kalman filter algorithm for sports video moving target tracking. In *2020 International Conference on Advance in Ambient Computing and Intelligence (ICAACI)*, pages 26–30, 2020.

[22] Hang Ran, Xin Ning, Weijun Li, Meilan Hao, and Prayag Tiwari. 3d human pose and shape estimation via de-occlusion multi-task learning. *Neurocomputing*, 548:126284, 2023.

[23] Oscar Real-Moreno, Julio C. Rodríguez Quiñonez, Wendy Flores Fuentes, Oleg Sergiyenko, Jesus E. Miranda Vega, Gabriel Trujillo Hernández, and Daniel Hernández Balbuena. Camera calibration method through multivariate quadratic regression for depth estimation on a stereo vision system. *Optics and Lasers in Engineering*, 174:107932, 2024.

[24] James G. Richards. The measurement of human motion: A comparison of commercially available systems. *Human Movement Science*, 18(5):589–602, 1999.

[25] Tomohiro Shimizu, Ryo Hachiuma, Hideo Saito, Takashi Yoshikawa, and Chonho Lee. Prediction of future shot direction using pose and position of tennis player. In *Proceedings of the 2nd International Workshop on Multimedia Content Analysis in Sports*, MMSports '19, page 59–66, New York, NY, USA, 2019. Association for Computing Machinery.

[26] Jin Sun, Heping Wang, and Xinglong Zhu. A fast underwater calibration method based on vanishing point optimization of two orthogonal parallel lines. *Measurement*, 178:109305, 2021.

[27] Jinbao Wang, Shujie Tan, Xiantong Zhen, Shuo Xu, Feng Zheng, Zhenyu He, and Ling Shao. Deep 3d human pose estimation: A review. *Computer Vision and Image Understanding*, 210:103225, 2021.

[28] Fritz Webering, Holger Blume, and Issam Allaham. Markerless camera-based vertical jump height measurement using openpose. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3863–3869, 2021.

[29] Tishya A.L. Wren, Pavel Isakov, and Susan A. Rethlefsen. Comparison of kinematics between theia markerless and conventional marker-based gait analysis in clinical patients. *Gait Posture*, 104:9–14, 2023.

[30] Haoran Yi, Deepu Rajan, and Liang-Tien Chia. Automatic extraction of motion trajectories in compressed sports videos. *Association for Computing Machinery*, page 312–315, 2004.