Master Thesis in ICT for Internet and Multimedia

Markerless Motion Analysis from Synchronized 2D Camera Views: A Convolutional Neural Network Approach

Supervisor
Michele Rossi
Università di Padova

Master Candidate
Francesco Piemontese

Co-supervisors
Matteo Gadaleta,
Simone Milani,
Zimi Sawacha

Padova, 7 October 2019
Academic Year 2018/2019

Università degli Studi di Padova

Department of Information Engineering

Master Thesis in ICT for Internet and Multimedia

Markerless Motion Analysis from Synchronized 2D Camera Views: A Convolutional Neural Network Approach

Supervisor
Michele Rossi
Università di Padova

Co-supervisors
Matteo Gadaleta,
Simone Milani,
Zimi Sawacha

Master Candidate
Francesco Piemontese

Academic Year 2018/2019

To Sole.

Abstract

Optical motion capture systems, which measure the movement performed by a subject from multi-view synchronized recordings, have been employed in biomedical research for the tasks of gait analysis and rehabilitation assessment. Although they achieve remarkable accuracy, they usually require controlled experimental conditions and a high number of cameras in order to function reliably.

On the other hand, recent keypoint detection approaches employing convolutional neural networks (CNNs) have proven capable of producing meaningful pose estimates from even a single monocular image, taken under general conditions. However, because the majority of these models target non-medical applications, their precision is generally not sufficient for use in a clinical setting.

In this work, we propose a CNN-based methodology for the clinically accurate tracking of a human subject. Making use of a state-of-the-art convolutional pose estimator, joint center locations were independently evaluated within the 2D views produced by synchronized calibrated cameras, and lifted to 3D space by means of triangulation. The precision of the resulting pose estimates was then increased through an original refinement routine, leveraging subject-specific kinematic information.

The performance of the developed pipeline was evaluated on a dataset made available by the University of Padova's BioMovLab, containing recordings of walking subjects taken by six synchronized cameras in both a controlled laboratory environment and a shallow swimming pool. Accurate joint angles trajectories were obtained for the hip and knee joints, highlighting the viability of the proposed approach.

Contents

Listing of figures

1
Introduction

Optical motion capture (MoCap) approaches measure the movement performed by a subject from the data collected by a set of synchronized cameras. In addition to their several applications in the fields of security and entertainment, they have been employed in biomedical research for the tasks of gait analysis and rehabilitation assessment.

Most commercially available MoCap systems employ a set of reflective markers, which are attached to the subject's skin and tracked by feature extraction algorithms throughout the considered video sequences. However, this procedure requires expensive, dedicated instrumentation (meaning it cannot be employed in the wild), and is reliant on the markers' visibility, causing it to fail in the presence of occlusions.

Due to these limitations, increasing attention has recently been devoted to the development of motion capture methodologies which do not rely on markers. These approaches, referred to as markerless, generally make use of subject-specific anatomical models, providing both morphological (3D shape) and kinematic information. Tracking is carried out by iteratively matching these descriptors to the data extracted from the considered recordings, via either background subtraction or visual hull reconstruction. Although these methods have achieved remarkable precision, they generally require controlled experimental conditions and a high number of cameras (≥ 6) in order to function reliably.

In a parallel line of work, impressive progress has recently been made in the field of pose estimation (PE): a branch of computer vision concerned with predicting, within images of human subjects, the location of anatomical keypoints such as the shoulders and knees. While early PE techniques employed handcrafted part-based descriptors, significantly improved performance has now been attained by models relying on convolutional neural networks (CNNs). However, because the majority of these approaches target non-clinical tasks (such as activity recognition and pedestrian detection), their accuracy is generally not sufficient for use in biomedical applications.

In this thesis, we explore the viability of a CNN-based gait analysis methodology, aimed at retrieving clinically meaningful movement information. Employing a state-of-the-art convolutional pose estimator, joint center locations were evaluated within each set of synchronized 2D views, and lifted into 3D space by means of triangulation. The accuracy of the resulting pose predictions was then increased through an original refinement routine, leveraging previously acquired anatomical data. Unlike most markerless motion capture approaches, we relied solely on kinematic information regarding the length of bone segments in the subjects of interest, and did not take morphological shape descriptors into consideration.

The performance of proposed approach was evaluated on a dataset made available by the University of Padova's BioMovLab, containing recordings of walking subjects taken by six synchronized cameras, in both a controlled laboratory environment and a shallow swimming pool.

The remainder of this thesis is structured as follows. In Chapter 2, we provide an overview of a state-of-the-art markerless motion capture approach, while contemporary CNN-based pose estimation techniques are illustrated in Chapter 3. Chapter 4 is dedicated to a description of the stages comprising the proposed methodology. The results obtained by our system are presented in Chapter 5 and reviewed, along with possible future developments, in Chapter 6.

2
Markerless Motion Capture

The term “motion capture”, or MoCap, refers to a wide range of computer vision techniques, concerned with the description and measurement of the movement performed by a subject. Originally developed as a photogrammetric tool in biomechanics research, motion capture has been studied extensively in recent years, due to the many applications exploiting it in the fields of medicine, entertainment and security.

In this work, we are particularly interested in the use of MoCap for clinical tasks, such as the analysis of human gait for rehabilitation assessment. Depending on their operating principle, systems developed for such purposes may be divided into two main categories: optical and non-optical. Within non-optical systems, movement-related parameters are directly measured by sensors (either electromagnetic or inertial) placed on the subject’s skin. Because of the encumbrance and weight of the wearable instrumentation, which significantly limits the practicable range of motion, this technology is rarely utilized in medical applications.

Optical methods on the other hand make use of the data collected by a set of synchronized calibrated cameras. In the most common approach of this kind, retroreflective or light-emitting markers are attached to the subject’s body, and tracked throughout the recorded sequences by means of feature-extraction algorithms [1]. The 3D trajectory of each marker is then triangulated from its 2D position in each frame, providing quantitative insight into the performed movement.

Due to their smaller size and weight, placing fewer limitations on the subjects’ movement compared to wearable sensors, the use of markers has seen widespread adoption in clinical tasks. Despite this, marker-based approaches remain affected by meaningful issues: firstly, in order for the triangulation process to be successful, each marker must be visible at all times to at least two cameras, leading to tracking failures in the presence of self-occlusions; furthermore, marker-placement is a time-consuming and partially non-repeatable process, requiring the presence of specialized personnel. Lastly, the presence of deformable soft tissue between markers and bones may result in non-negligible measurement errors.

In light of these limitations, growing attention has recently been devoted to the development of MoCap methodologies which do not rely on markers. Although the accuracy of such approaches is not yet on par with that of marker-based ones, promising results for the tracking of lower-body limbs have been shown in studies such as [2] and [3]. A description of the markerless motion capture (MMC) pipeline employed in these works is provided in the following sections.

2.1 Visual Hull Generation

As previously discussed, optical motion capture approaches make use of the data collected by a set of synchronized calibrated cameras, recording a subject's movement from multiple viewpoints. While in marker-based systems the resulting videos may be processed directly, markerless methodologies generally require a background-subtraction preprocessing step, highlighting the area occupied by the subject of interest throughout the considered movement sequence.

Due to its widespread use in computer vision applications, background subtraction has been researched extensively in recent years, leading to the development of several dedicated techniques. Previous studies testing the efficacy of the MMC pipeline [2, 3] rely on the approach presented in [4], in which the colour of each background pixel is modelled by an iteratively updated Gaussian mixture. More recently, superior performance was achieved by means of deep learning based models, a review of which may be found in [5]. Regardless of the chosen implementation, only the information pertaining to the subject's position in the scene is retained for further processing.

The binarized images obtained from each set of synchronized views (referred to as silhouettes) are used to derive a 3D representation of the subject, according to

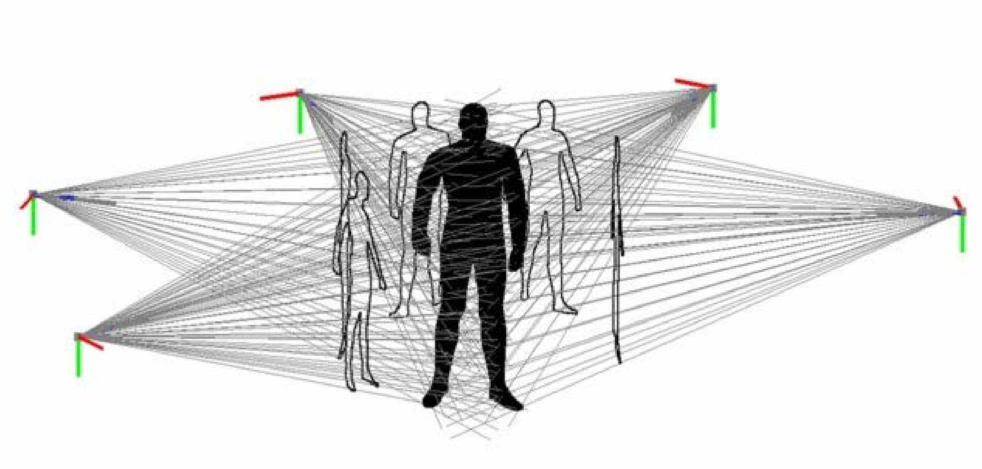


Figure 2.1: Visual hull reconstruction from silhouettes (image from [7]).

the procedure displayed in Figure 2.1. Since the position and orientation of each camera are known, each silhouette can be back-projected into the observed space, in a generalized cone whose cross-section depends on silhouettes. The intersection of these cones denotes a locally convex over-approximation of the volume occupied by the subject, known as a visual hull (VH) [6]. The subject's movement throughout a video may then be represented using a sequence of visual hulls, one for each frame.

Due to the difficulty of representing complex 3D shapes, an approximate VH construction procedure, referred to as volume carving [8], is often employed in practice. The calibrated space is partitioned into cubic volumes known as voxels, each of which is considered part of the visual hull if and only if its projection on the image plane of each camera belongs to the corresponding silhouette. The resolution of the computed VH decreases with the size of the voxels.

It should be noted that the visual hull construction procedure may sometimes yield rather inaccurate results; in particular, concavities and self-occlusions may lead to the creation of phantom volumes: artefacts occurring when an area of the surface of interest is simultaneously blocked from the view of all cameras. The number of employed viewpoints and their placement around the scene have been shown to be instrumental in the alleviation of this problem [9].

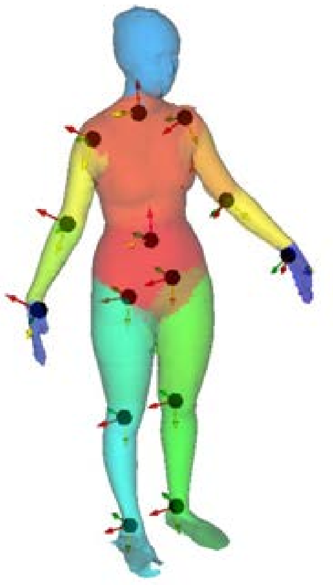


Figure 2.2: Example of a segmented anatomical model, in which each color represents a different body part and joint centers are denoted by black circles (image from [2]).

2.2 Model Definition

A sequence of visual hulls alone is not sufficient for the accurate tracking of a human subject. Because the procedure described above is carried out independently for each set of synchronized frames, no correspondence is established between VH points generated in consecutive time instants. Moreover, the computed meshes are not segmented, meaning they contain no information about which vertices belong to which body part. For these reasons, visual hulls are employed in the MMC pipeline as a noisy guide, to which an accurate anatomical model is matched. Such a model encodes subject-specific morphological (3D shape) and kinematic (joint centers location) information, and is comprised of two main components:

- a triangular mesh, obtained through visual hull construction or laser scan;
- an articulated model of the human body, consisting of a set of rigid segments connected in a tree-like structure by spherical joints with six degrees of freedom.

Depending on the target application, different articulated models may be defined;

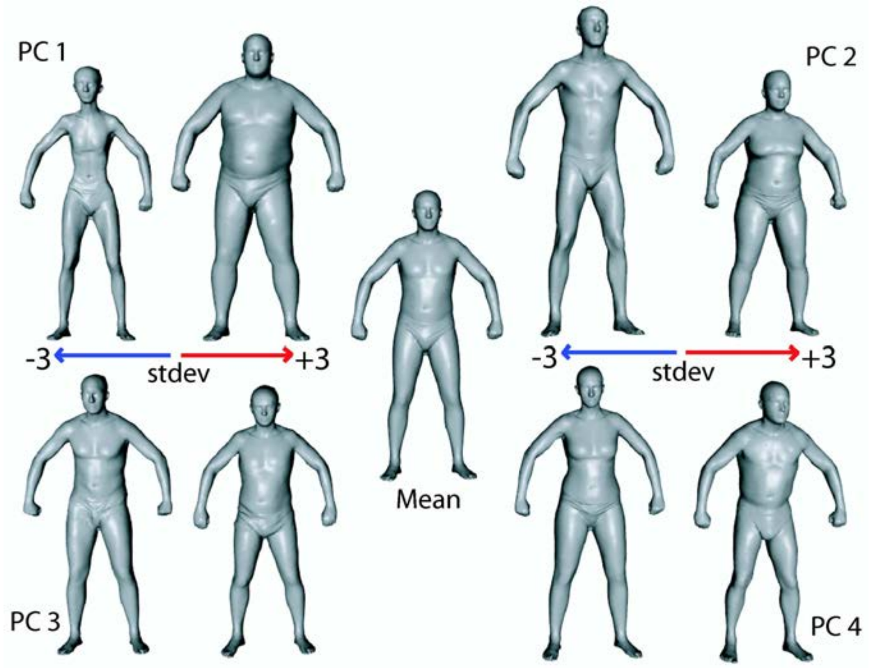


Figure 2.3: Body shapes obtained by modifying the SHS's first four principal components (image from [11]).

the one employed in [2], shown in Figure 2.2, uses of 15 segments[1] and 14 joints.

Model generation is carried out according to the procedure presented by Corazza et al. in [10], which automatically performs mesh segmentation and joint centers localization. The proposed processing pipeline makes use of two algorithms:

- a pose registration algorithm (described in detail in Section 2.3), which for each segment of an articulated model computes the rigid transformation necessary to match the pose of a target mesh;
- a shape registration algorithm, which given two meshes (assumed to be in a similar reference pose) determines the deformation required to morph the former into the latter.

The latter is based on the previous work carried out in [11]. From a dataset of laser scans of human subjects in a similar reference pose, the authors developed a space of human shapes (SHS), the dimensionality of which was reduced by means of principal component analysis (PCA, [12]). As shown in Figure 2.3, this provides a low-dimensional representation of highly complex body shapes, allowing the arguments

[1] Pelvis, torso, head, forearms, arms, hands, thighs, shanks and feet.

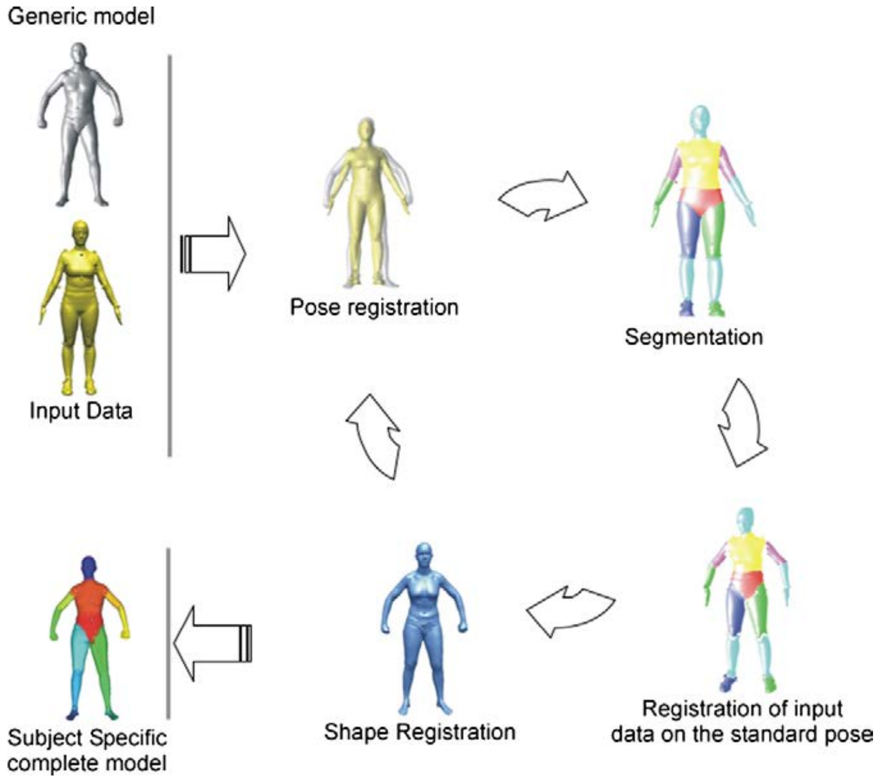

Generic model
Input Data
Pose registration
Segmentation
Registration of input data on the standard pose
Shape Registration
Subject Specific complete model

Figure 2.4: Automatic model generation pipeline (image from [10]).

of the shape registration algorithm to be expressed by a linear combination of few[2] SHS principal components.

The two algorithms discussed above are combined into a pose-shape registration procedure (whose steps are summarized in Figure 2.4), which from a manually segmented reference mesh R (the center of the space of human shapes) performs segmentation of a subject data mesh S by iterating the following steps:

1. Using the pose registration algorithm, determine the configuration of the reference mesh which best matches the subject mesh.

2. Segment the subject mesh by associating to each point $p \in S$ the same rigid body segment as the closest (in terms of Euclidean distance) point $q \in R$.

3. Register the subject mesh in reference pose, applying to it the inverse of the transformation found in step 1.

[2] In [10], the first 10 principal components of the space are used.

4. Replace R with the SHS model, determined using the shape registration algorithm, which most closely resembles the transformed subject mesh.

The process is iterated until convergence, which occurs when there is no difference between the subject and reference meshes (the authors of [10] mention that four to six iterations are usually sufficient). Note that the transformation applied to S in step 3 is necessary for the application of the shape registration algorithm, as all models in space of human shapes are in reference pose.

Once the segmented subject mesh has been computed, the position of each joint center is derived as a linear combination of seven vertices in its vicinity. For all joints but the hips, the error associated with this procedure was shown in [10] to be comparable to that caused by marker misplacement in marker-based methods.

Linking kinematic information to reliable morphological features allows the generation of a subject-specific anatomical model to be carried out in a fully automatic manner, removing one of the potentially most time consuming steps of the MMC pipeline. Furthermore, the proposed methodology can work (albeit with slightly lower accuracy) with 3D shape data acquired from visual hull construction rather than laser scans, the required instrumentation for which may not be available.

2.3 Model Matching with Articulated ICP

Articulated ICP [13] is a generalization to articulated models of the standard ICP algorithm [14, 15], commonly employed for the registration of 3D surfaces. Consider a visual hull V, consisting of a point cloud $Z = \{z_1, \ldots, z_K\}$, and a model Y defined by:

- a set $X = \{x_1, \ldots, x_M\}$ of surface points;
- a set $P = \{p_1, \ldots, p_N\}$ of rigid segments;
- a set Q of joints connecting the segments.

The algorithm's goal is to match the model to the VH, aligning the surfaces X and Z via a set of rigid transformations $T = \{T_1, \ldots, T_N\}$ (one per segment), while respecting the anatomical constraints placed on Y by the segments' rigidity and joints' position. This is done by iteratively performing the following two steps:

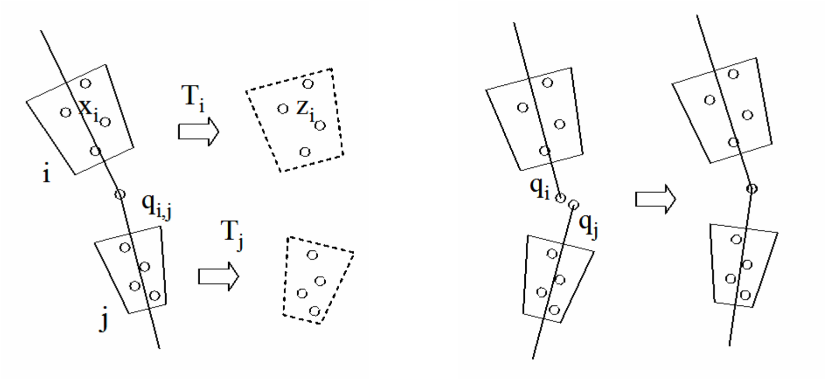


Figure 2.5: Left, pairs of corresponding points on the model and VH meshes, whose distance is penalized by *H(T)*. Right, alignment of joints induced by *G(T)* (images from [13]).

1. For each $x_i \in X$, determine the corresponding $z_{c[i]} \in Z$, where $c\colon X \to Z$ is an injective function associating to each point on the model surface the closest (in terms of Euclidean distance) point on the VH mesh.

2. Given a set of corresponding points $\{(x_i, z_{c[i]})\}_{i=1}^{M}$, find and apply the set of transformations which minimizes the non-linear energy function (see below for the definition of the constituting terms):

$$F(T) = G(T) + H(T). \tag{2.1}$$

   This task is carried out using the Levenberg-Marquardt curve-fitting algorithm [16], which provides a good tradeoff between accuracy and speed.

The first term in (2.1) enforces the correct alignment of the model and visual hull meshes. Specifying each roto-translation $T_j \in T$ by means of two vectors r_j, $t_j \in \mathbb{R}^3$ (twist and translation parameters, respectively), and letting $R(r_j)$ denote the rotation matrix induced by r_j:

$$H(T) = \sum_{i=1}^{M} \left\| R(r_{l[i]})x_i + t_{l[i]} - z_{c[i]} \right\|^2, \tag{2.2}$$

where $l\colon \{1, \ldots, M\} \to \{1, \ldots, N\}$ maps the index of a point on the model surface to that of the segment it belongs to.

The second component of the energy function introduces soft constraints on the position of the joints, which prevent the model from reaching anatomically incorrect configurations. For each pair $(p_i,\, p_j)$ of adjacent segments, the transformations T_i and T_j are forced to predict approximately the same location for the joint that connects them:

$$G(T) = w_G \sum_{(i,j)\in Q} \left\| R(r_{l[i]})q_{i,j} + t_{l[i]} - R(r_{l[j]})q_{i,j} - t_{l[j]} \right\|^2 . \tag{2.3}$$

The term w_G regulates the relative importance of the two loss contributions. By increasing it, a greater joint consistency is enforced, at the cost of a generally less accurate matching between the two point clouds.

3
Pose Estimation

Human pose estimation (PE) is the problem of determining the body configuration of human subjects appearing in images and videos [18]. Commonly, this is achieved by evaluating the position of a set of anatomical keypoints (such as elbows, shoulders and knees), defining a skeleton-like model for the body via pairwise connections (referred to as limbs or bones) of these joints. The task of pose estimation (of which an example is shown in Figure 3.1), is pivotal to several computer vision applications, including:

- **Autonomous driving**: pedestrian detection and accident avoidance;
- **Automated surveillance**: tracking of a subject, person counting, detection of suspicious activities;
- **Activity recognition**: sign language understanding, fall detection and prevention, human-computer interaction;
- **Entertainment**: animation, CGI, augmented reality;
- **Markerless motion capture** for clinical assessment and rehabilitation.

Because of its wide range of uses, pose estimation has been extensively researched over the last two decades, leading to the development of remarkably accurate models [19–23]. Despite this, several aspects of the problem remain challenging to this

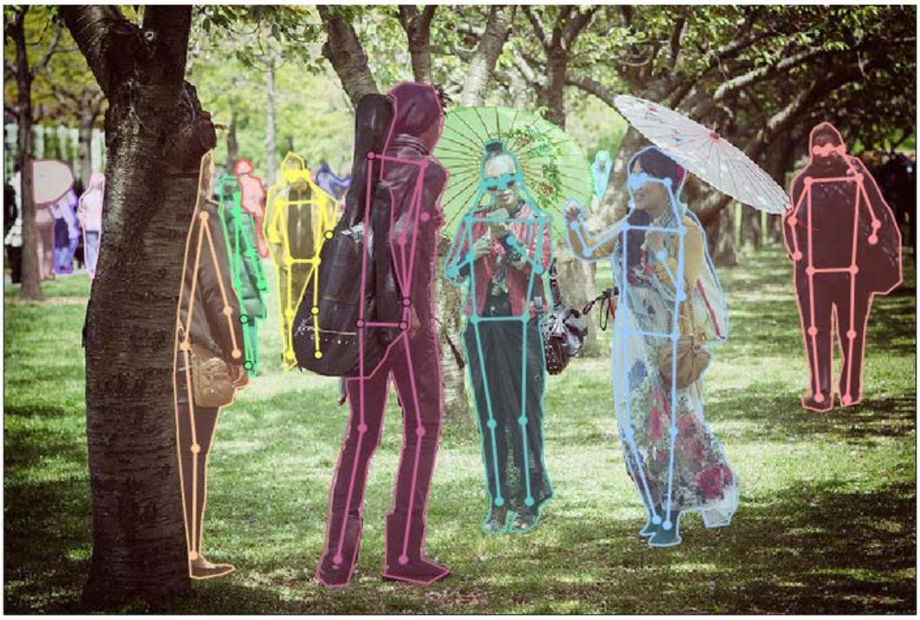


Figure 3.1: An example of pose estimation (image from the COCO dataset [17] website).

day. The main difficulty faced by PE approaches is the huge variability of human appearance in images. On top of the sheer number of possible body configurations, differences in orientation, physique, clothing and lighting conditions have to be taken into account. Furthermore, not all joints may be visible at the same time, due to occlusions by other body parts or external objects. The combination of these factors results in an extremely large search space.

In the following, we provide an overview of the PE approaches proposed in the last years, while a detailed description of the model employed in this work is given in Section 4.2.

3.1 Pictorial Structures

As previously mentioned, pose estimation has been a widely studied topic in computer vision for over 15 years. Prior to the advent of deep learning, most models relied on the Pictorial Structures (PSs) framework, originally introduced in [24] for the task of object detection (particularly human faces) and later popularized for PE by the authors of [25]. In this approach, the object of interest is described by a set of local descriptors (referred to as parts), arranged in a deformable configuration by means

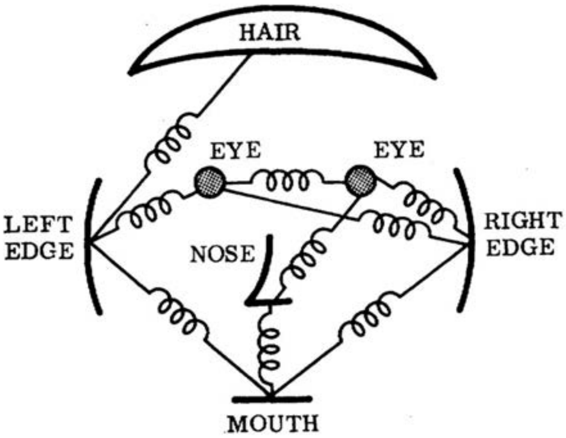


Figure 3.2: Example of a pictorial structures model for face detection (image from [24]).

of pairwise connections, often visualized as springs (see Figure 3.2 for an example relative to face detection). In the context of pose estimation, parts represent body elements such as arms and legs, with the links between them modelling articulations.

Mathematically, an n-part pictorial structure may be described by means of an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, in which each node in $\mathcal{V} = \{v_1, \ldots, v_n\}$ corresponds to a part, and the edge (v_i, v_j) exists in $\mathcal{E}$ if and only if v_i and v_j are connected. If $\mathcal{G}$ is fixed, the PS's configuration may be characterized by specifying the location (combination of position, orientation and foreshortening) of each of its parts. Using the same notation as [25], let:

- $m_i(l_i)$ quantify the mismatch between the image and part v_i in location l_i;
- $d_{ij}(l_i, l_j)$ measure the deformation resulting from parts v_i and v_j, such that $(v_i, v_j) \in \mathcal{E}$, being in locations l_i and l_j respectively.

Then, the problem of pose estimation may be reformulated in terms of the minimization of an energy function, i.e. finding the configuration L^* such that:

$$L^* = \underset{P = \{l_1, \ldots, l_n\}}{\arg\min} \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i, v_j) \in E} d_{ij}(l_i, l_j) \right). \tag{3.1}$$

Although part-based models offer an intuitive way to represent articulated bodies, approaches relying on the pictorial structures framework have often struggled to

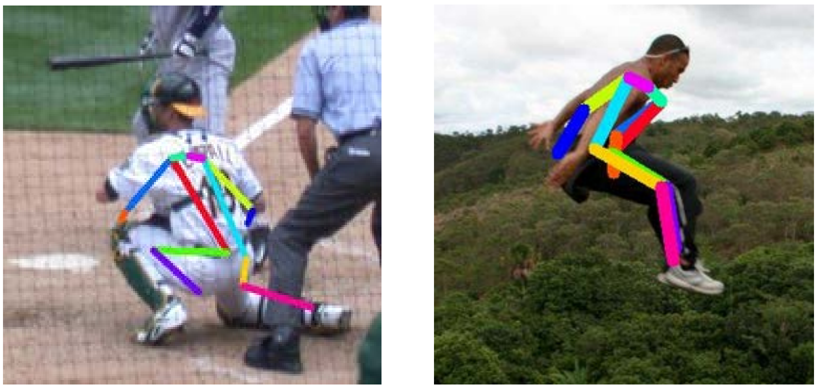


Figure 3.3: Example of problematic poses for pictorial structures (images from [26]).

correctly predict complex poses (such as the ones shown in Figure 3.3), in which multiple joints are occluded. The cause of this limitation is the inability of PS methods to capture global context in images: predictions are based on local part descriptors, with only few of the interactions between parts being taken into account (usually only first-degree connections between directly linked parts). Furthermore, rather than being inferred from images, these interactions have to be manually selected before the model is run, severely limiting its expressiveness. Several works attempted to address these issues, either by employing more effective part detectors [27, 28] or by modelling more complex dependencies among parts [29, 30]. Although these models did achieve notable results, their performance has since been surpassed by approaches employing deep learning techniques. As discussed in Section 3.2, all recent pose estimation systems are based on convolutional neural networks.

3.2 Convolutional Neural Networks

3.2.1 DeepPose

The first work to apply deep learning techniques to the task of pose estimation was presented by Toshev and and Szegedy [26] in 2014. They employed a convolutional neural network (CNN or ConvNet), referred to as DeepPose, to predict the position of a set of anatomical keypoints, performing regression over single-person views.

Compared to the models described in Section 3.1, this approach has the advantage of being able to produce significantly more general representations. While PS-based

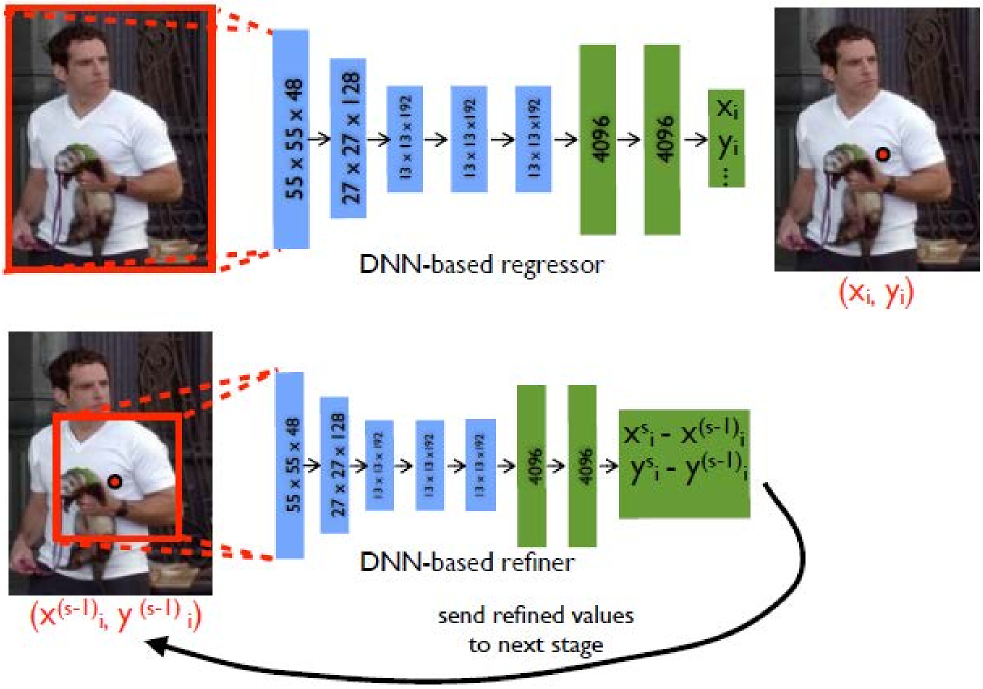


Figure 3.4: Top: the architecture of DeepPose's main CNN module. Bottom: refinement of the predictions produced by stage $s-1$, carried out in stage s. Blue rectangles denote convolutional layers, while green ones denote fully connected ones. Images from [26].

approaches formulate predictions by matching a local part template to areas of an image, CNNs take the whole image as input, computing features at multiple different resolutions. This allows them to reason about pose in a holistic manner, capturing the global context of joints to an extent that pictorial structures cannot. Furthermore, the use of a CNN removes the need to explicitly design part descriptors and the interactions between them, which are instead implicitly learned by the network.

The architecture of the proposed network's main module is shown in Figure 3.4 (top). The input image, having a resolution of 220×220, is downsampled to 55×55 using a convolutional filter with a stride of 4, and passed to a ConvNet consisting of 5 convolutional layers (shown in blue), and 2 fully connected ones (shown in green). The size of the filter banks is 11×11 and 5×5 for the first two convolutional layers, and 3×3 for the remaining three. The representations' resolution is decreased using max-pooling, and ReLU [31] activation is employed after all layers (both convolutional and fully connected).

The output of the network is a set of 2D coordinates, estimating the position of k

joints (14 for full-body views, 10 if only the upper body is shown). Due to the low input resolution (necessary to maintain a manageable computational complexity) a single iteration of this procedure is unable to generate sufficiently accurate predictions. The authors address this by running the input images through a cascade of three convolutional pose regressors (referred to as modules in [26]), each employing the same, previously described architecture. As shown in Figure 3.4 (bottom), modules following the first one take as input k bounding boxes centered on the previous stage's outputs, and are trained to estimate the offset between these predictions and the true joint locations.

The CNN-based system presented in [26] was able to achieve state of the art results on the FLIC and Leeds Sports datasets [32, 33]. Moreover, it did so using a relatively simple model, very similar in structure to the one used for image classification in [34]. The potential of deep learning showcased by this work caused the focus of pose estimation research to shift entirely from pictorial structures to ConvNets, with several architectures proposed in the span of just a few years. An overview of contemporary pose estimation approaches is given in the next sections.

3.2.2 Heatmap Regression

The proliferation of CNN-based PE models brought about by DeepPose resulted in several modifications being made to Toshev and Szegedy's original design. The first, and perhaps more significant, concerns the nature of the network's output. In [26], this consists of the (x, y) coordinates of the k joints, which are regressed to directly using fully connected layers. Tompson et al. however argued [35] that this may lead to imprecise predictions when the position of a joint is affected by uncertainty.

This problem is made worse by the use of pooling. All PE systems relying on ConvNets use strided pooling layers to (1) reduce the model's computational complexity and (2) introduce invariance to local input transformations. However, by reducing the resolution of feature maps, these operations also decrease localization accuracy.

The authors of [35] employ an alternative network structure which, rather than image coordinates, produces a set of k heatmaps $\{\mathcal{H}_1, \ldots, \mathcal{H}_k\}$. As shown (for a subset of the joints) in Figure 3.5, for each $i \in \{1, \ldots, k\}$ $\mathcal{H}_i$ describes the per-pixel likelihood of finding the i^{th} joint in each output position.

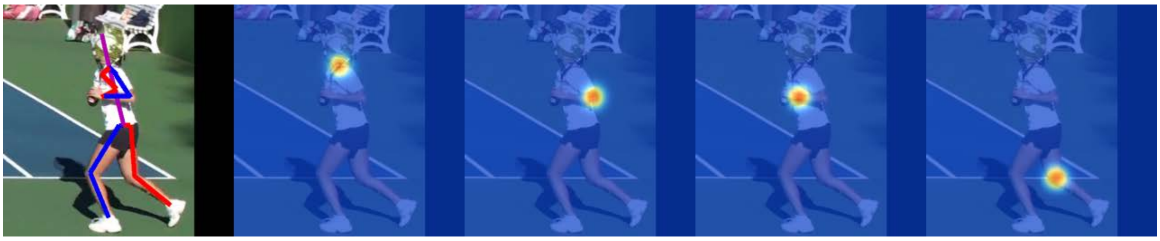


Figure 3.5: Example heatmaps produced by a CNN-based pose estimator for a subset of joints (image from [36]).

The proposed architecture is displayed in Figure 3.6. A 3-level[1] Gaussian pyramid [37] obtained from the input image is passed to a "Coarse Model" (CM) to obtain a set of low resolution heatmaps. The highest-valued coordinates of each heatmap yield a rough estimate for the joints' position, which are used to crop regions of interest from the first two feature maps of the CM. These are in turn run through a "Fine Model" (FM), which refines predictions and returns higher resolution heatmaps. For both the CM and FM, the training loss is defined as the SSE (across all joints) between the predicted and ground-truth heatmaps, the latter being 2D Gaussians of small variance centered at the annotated ground-truth positions. The cascade of the two models is trained by minimizing a combination of the two losses.

The model proposed by Tompson et al. achieved state of the art results on the FLIC and MPII [27] datasets. More than to their two-stage architecture, this success can be attributed to their use of heatmaps, which enables the CNN to (1) express a measure of confidence in its predictions and (2) deal with multi-modal outputs (where multiple locations are considered possible for the same joint). This is made apparent

[1] Only two levels are shown in Figure 3.6.

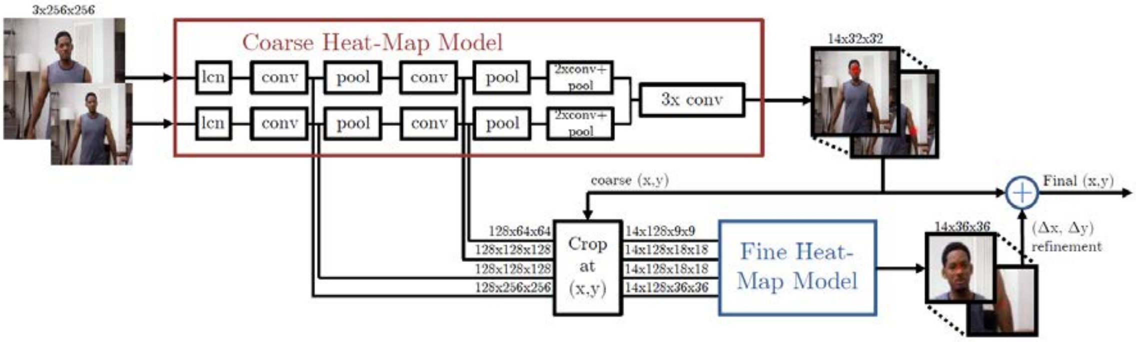


Figure 3.6: Architecture of the CNN proposed by Tompson et al. (image from [35]).

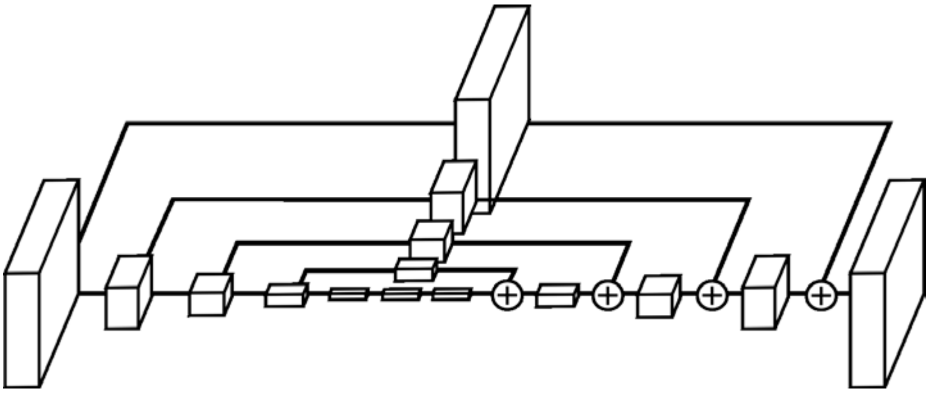


Figure 3.7: Architecture of an hourglass module (image from [36]).

by the fact that, while their coarse-to-fine pipeline was eventually surpassed, their heatmap description is still widely in use.

3.2.3 Stacked Hourglass Networks

Perhaps the most influential work to employ the heatmap output representation is "Stacked Hourglass Networks" [36], a paper presented by Newell et al. in 2016, which laid the foundations for several contemporary systems. The authors argue that, in order to accurately perform PE, a model must be able to leverage information from different scales: while low-level features are required to detect highly recognizable attributes, such as a human hand or face, a higher level understanding is necessary to capture orientation and the relations between limbs. For this purpose, they propose a novel CNN architecture, which they dub "hourglass" due to the shape of its representation, shown in Figure 3.7.

The resolution of the input image (initially 256 × 256) is progressively reduced, using a sequence of convolutional and max-pooling layers, down to 4 × 4 (bottom-up processing). Skip connections (similar to the ones employed in previous image segmentation approaches [38, 39]), introduced before each pooling step, are used to preserve spatial information across scales, and to apply further convolutions to the computed feature banks. Once the minimum resolution is reached, the network recovers the original input size by means of consecutive nearest-neighbour upsampling operations (top-down processing). The second half of the hourglass mirrors the first one in structure (meaning the number and shape of the feature banks are the same),

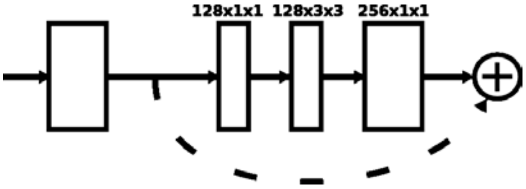


Figure 3.8: Structure of a residual module employed in Stacked Hourglass Networks (image from [36]).

and after each upsampling step element-wise addition is performed between the computed representations and the ones carried by skip connections from the corresponding bottom-up stage. Through the combination of these procedures, the network is able to compute image features at different resolutions, and to then bring them together in order to understand global pose context. The final prediction heatmaps (one per joint) are obtained from the last stage by applying two additional 1×1 convolutional filters[2]. Note that each box in Figure 3.7 represents a residual module such as the one shown in Figure 3.8. Such modules have seen widespread adoption in recent deep learning applications, as they were shown [40] to greatly facilitate the training of deep networks.

With an approach similar to [26], the authors concatenate eight of the hourglass modules described above (hence the name “Stacked Hourglass Networks”), resulting in the architecture displayed in Figure 3.9. The input of each module is the element-wise sum of the previous one’s final feature representations and prediction heatmaps (to which 1×1 convolutions are applied to match the number of channels). Importantly, intermediate supervision is employed, meaning a loss (the same MSE loss employed in [35]) is applied to the heatmaps produced by all stages rather than just the final one.

[2] Not shown in Figure 3.7

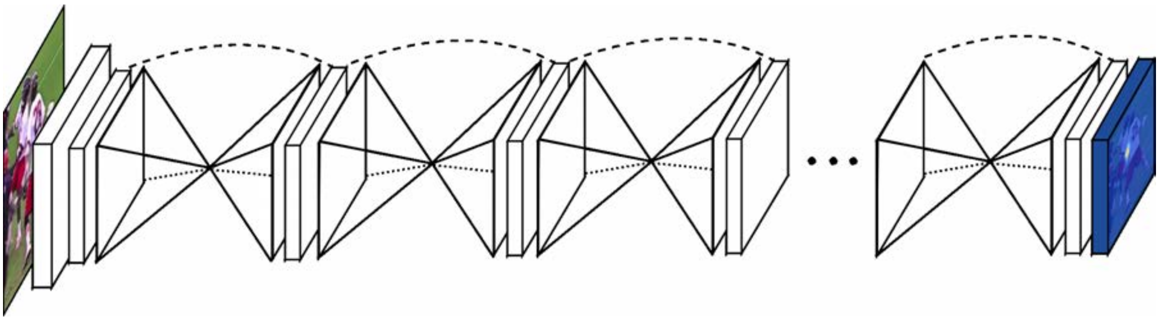


Figure 3.9: Architecture of a Stacked Hourglass network (image from [36]).

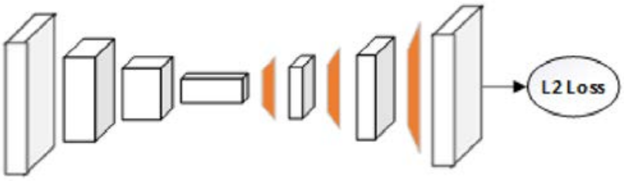


Figure 3.10: Architecture of the Simple Baselines network (image from [20]). The orange rectangles represent deconvolutional modules.

This forces the network to acquire a comprehensive understanding of the input image before a forward pass is complete, allowing subsequent modules to either refine or reassess intermediate predictions through repeated bottom-up, top-down inference.

The CNN proposed by Newell et al. was able to outperform all previous approaches on the FLIC and MPII datasets, and is still competitive today[3]. Note that, while in [42] and [35] a part-based model is employed on top of ConvNets to enforce pose consistency, this result was achieved without any explicit representation of the human body. Moreover, the practice of purposely designing the CNN pipeline to capture image features at different scales was widely adopted by later models, including the one [19] employed in this thesis (see Section 4.2 for details).

3.2.4 Simple Baselines

It is worth mentioning an elegant work [20] by Xiao et al., striking in both its effectiveness and simplicity.

The proposed architecture, shown in Figure 3.10, is referred to as Simple Baselines. It largely consists of a ResNet backbone (commonly employed for the extraction of image features, [40]), with the only addition being three transposed convolution modules, represented above by orange rectangles. Originally introduced in [43], these layers (also known as deconvolutional) allow for upsampling operations whose parameters, rather than being pre-determined, are learned at training time (see [44, Chapter 4] for a detailed description).

This network structure is reminiscent of Stacked Hourglass: the input images undergo repeated downsampling operations, after which the original resolution is recovered to compute the final (heatmap-based) predictions. However in [36], because the

[3] A Torch7 [41] implementation of the network is available at http://www-personal.umich.edu/~alnewell/pose

upsampling process is carried out with a simple nearest-neighbour approach, convolutional skip connections had to be introduced to preserve spatial information across stages. By combining upsampling and convolutions in deconvolutional modules, the need for this expedient is lifted in [20], resulting in a comparatively much simpler architecture. Despite this, Simple Baselines was able to achieve state-of-the-art results on both the COCO and MPII datasets, significantly improving on the performance of Stacked Hourglass. A following approach by the same authors, known as High-Resolution Net [19], is employed for pose estimation in this thesis, and described in detail in Section 4.2.

3.2.5 Considerations

In this section, we provide a few closing remarks regarding pose estimation strategies alternative to the one employed in this thesis, as well as insight into why we chose not to pursue them.

Contemporary 2D PE systems can be divided into two main groups: top-down and bottom-up. In top-down approaches, the network input is assumed to be a view of a single person (or at least an image in which one individual is prominent compared to the others). All the architectures described in the previous sections fall into this category; in order for them to work in a multi-person environment, a person detector has to be employed beforehand, to draw around each subject in the scene a bounding box on which PE can be carried out. This means that if person detection fails, pose estimation will fail as well. Moreover, because the model has to be run once for each detected bounding box, the overall runtime will be proportional to the number of people in the image.

Bottom-up approaches on the other hand operate by detecting body parts first, associating them into coherent poses at a later stage. Notable examples of this paradigm are DeepCut [45], DeeperCut [46], OpenPose [23] and PifPaf [47]. Bottom-up approaches have two main advantages over top-down ones:

1. their performance is not tied to that of a person detector;
2. their runtime is independent of the number of people in the scene, as body parts are detected in parallel and can be aggregated efficiently.

These however are not relevant for the purposes of this thesis. Because our target application is clinical in nature, it is reasonable to assume a controlled environment, in which (1) a person detector is unlikely to fail and (2) only one person will be in view at a given time. As the perks of bottom-up methods come at the cost of a lower average precision [48], we decided against their use in this work, and to instead focus on more accurate top-down approaches.

To conclude this section, we briefly discuss current 3D pose estimation strategies, and the way in which they differ from the one employed in this work. Driven by applications such as animation and action recognition, several works addressing this task were presented in recent years; however rather than from multiple views, the majority of them recover 3D pose information form a single monocular image [49–52]. The reason for this lies mostly in the lack of comprehensive annotated 3D pose data: existing datasets [53, 54] were built using marker-based motion capture strategies, which require dedicated instrumentation and thus cannot be performed “in the wild”. To compensate for this deficiency, many authors attempted to leverage the extensive datasets available to single-person 2D pose estimators, with models that first formulate a 2D prediction, and then “lift” it to 3D. Because a multi-camera setup was available to us, we did not need to resort to these strategies.

CNN-based 3D PE approaches which employ multiple calibrated cameras are fairly rare in the literature, again primarily due to the absence of adequate datasets. In [55], differently oriented views of a subject are used in the training of an autoencoder to capture a latent 3D human body representation, however the pose estimator which makes use of this information only takes one image as input. The system developed by Carraro et al. [56] uses asynchronous Kinect cameras, and produces a 3D pose estimate by combining 2D predictions with the sensor depth information.

The approach most similar to ours is perhaps the one presented in [57]. The authors independently carry out 2D pose estimation on multiple camera views, and then recover the 3D position of joints by means of triangulation. In order to enforce a human body structural prior, a 3D pictorial structures (3DPS [58]) model is additionally employed. In trying to reduce the noise resulting from triangulation, this step serves a similar purpose to our prediction-refinement procedure, described in Section 4.4. We do not however rely on the PS framework; instead, we leverage previously acquired anatomical information to enforce subject-specific constraints on the joints.

4
Methodology

The main goal of this work is to investigate the viability of a CNN-based approach for the clinically accurate tracking of a human subject, providing a research direction alternative to the established MMC pipeline. For this purpose, we developed an original motion capture methodology (a summary of which is provided in Figure 4.1), structured into five main steps:

1. **Acquisition** (Section 4.1): an array of n synchronized calibrated cameras[1] is employed to record a walking human subject from multiple viewpoints.

2. **2D pose estimation** (Section 4.2): the recordings acquired in step 1 are passed as input to a convolutional pose estimator, referred to as High-Resolution Net [19]. For each frame of each considered sequence, the model returns 17 sets of 2D coordinates (estimating the position of key joints such as the elbows and knees), each of which is associated with a confidence score, expressing the likelihood of the prediction being correct.

3. **Triangulation** (Section 4.3): the 2D predictions made for a joint's location across the n synchronized views are used to triangulate its 3D position in the calibrated space. In order to prevent spurious detections from compromising the triangulation process, the confidence scores associated with each estimate are also taken into account in this phase.

[1] Our application supports any $n \geq 2$, though a greater number of cameras is recommended for increased accuracy.

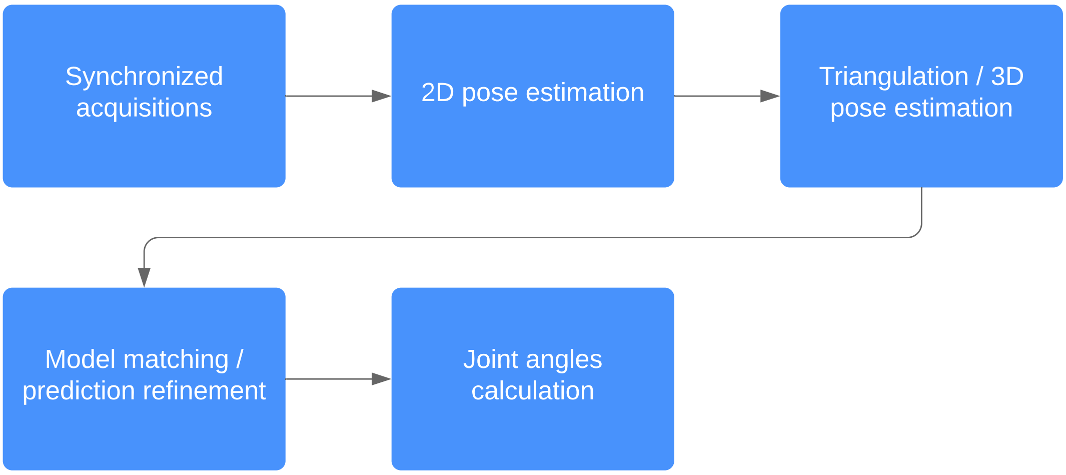

Synchronized acquisitions
2D pose estimation
Triangulation / 3D pose estimation
Model matching / prediction refinement
Joint angles calculation

Figure 4.1: Flowchart summarizing the proposed motion capture pipeline.

4. **Model matching** (Section 4.4): in order to preserve kinematic information collected in [2], concerning the length of rigid bone segments in the subjects of interest, the output of step 3 is refined by means of an original two-step optimization procedure, matching to each 3D pose estimate a subject-specific articulated model.

5. **Joint angles calculation** (Section 4.5): an original procedure for the definition of joint reference systems is applied to the pose predictions obtained in step 4. Its output is then employed to compute the joint angles associated with the hips and knees, characterizing the motion of the subject's lower limbs throughout the synchronized video sequences.

Although tracking approaches exploiting multiple synchronized cameras have been proposed in the literature (see Section 3.2.5), they were generally designed for non-clinical applications such as pedestrian detection and activity recognition, which only require approximate joint locations and a high-level understanding of the subject's pose. In this work, a greater emphasis was placed on the retrieval of clinically accurate joint angles, through the incorporation of previously acquired subject-specific information.

Unlike most model-based MMC approaches, relying on both morphological and kinematic subject descriptors, we did not employ any volumetric data. Rather than being placed on a 3D mesh, each joint's position was independently evaluated within each set of synchronized 2D views, and lifted to 3D space by means of triangulation.

A detailed description of the stages comprising the developed pipeline is provided in the following sections. Note however that the proposed methodology is highly modular, meaning changes could be made to the implementation of any of its components with minimal modifications to the others.

4.1 Experimental Setup

In order to meaningfully compare the results produced by our methodology and by the MMC pipeline, we evaluate our system's performance on recordings which in previous studies [2, 3] were processed using markerless motion capture techniques.

Three healthy male subjects were asked to perform six walking trials at self-selected speed in two different settings: a shallow swimming pool, with water up to their shoulders, and a controlled laboratory environment. Acquisitions were carried out using six synchronized underwater cameras (TS-6021PSC, Tracer Technology Co. Ltd), whose position around the volume of interest (shown for both experiments in Figure 4.2) was chosen in such a way as to minimize the number of simultaneous self-occlusions.

Camera calibration was carried out by means of the algorithm[2] presented by Bouget in [59]. The technique makes use of the widely adopted pinhole camera model; in its ideal formulation (not including lens distortion), this description expresses the relation between a 3D point $P = (x, y, z)$ in the camera's reference frame and its projection $p = (u, v)$ onto the image plane as:

$$p = \frac{1}{z} \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \end{bmatrix} P, \tag{4.1}$$

where:

- f_x and f_y denote the vertical and horizontal focal lengths (the distances between the center of projection and the image plane) in pixel units;
- $c = (c_x,\, c_y)$, referred to as principal point, denotes the projection onto the image plane of the focal point;

[2] We refer the reader to [2, Section 3.2.3] for a detailed description of the equipment employed to perform calibration in an both an underwater and laboratory environment.

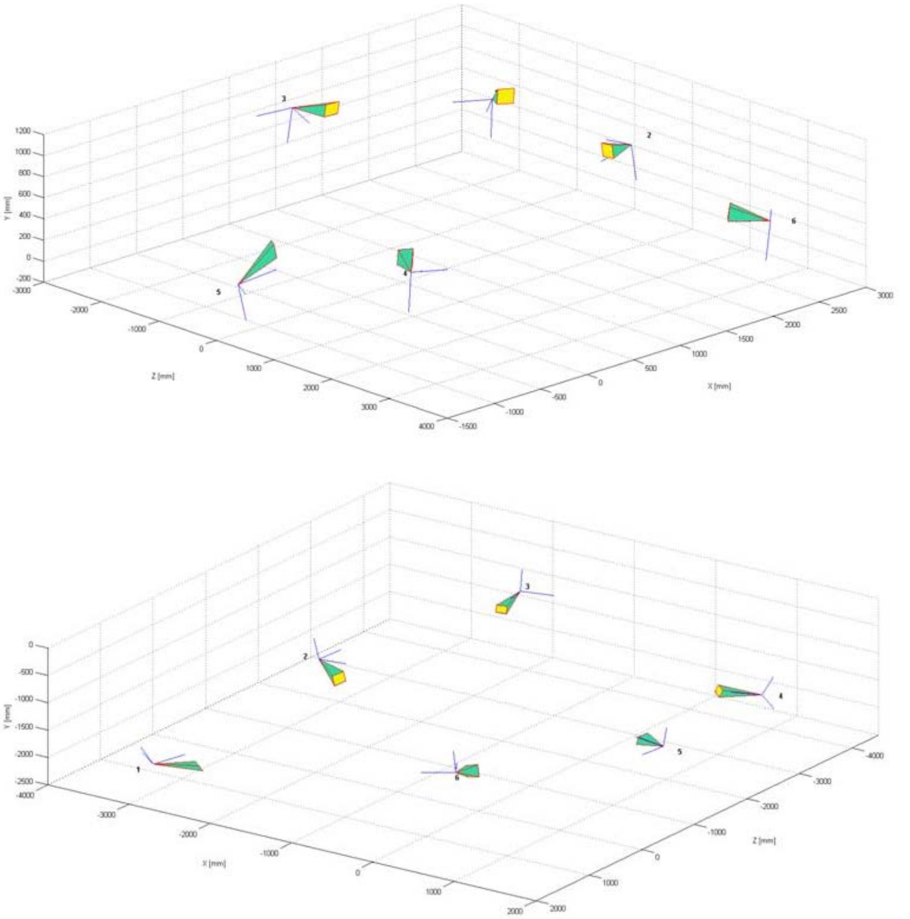


Figure 4.2: Camera configurations (as resulting from extrinsic calibration) employed to record the out-of-water (top) and underwater (bottom) walking trials.

The focal length and principal point are known as intrinsic parameters, and do not depend on the camera orientation or scene viewed.

Letting $r^2 = (x^2 + y^2)/z^2$, lens distortion may be accounted for by introducing an additive and a multiplicative term in equation (4.1), obtaining:

$$p = \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{z} \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \end{bmatrix} P + \frac{1}{z} \begin{bmatrix} 2p_1 xy + p_2(r^2 + 2x^2) \\ p_1(r^2 + 2y^2) + 2p_2 xy \end{bmatrix}, \qquad (4.2)$$

where the parameter vectors (k_1, k_2, k_3) and (p_1, p_2) characterize radial and tangential distortion, respectively. Through an iterative approach, Bouget's algorithm estimates the intrinsic parameters and distortion coefficients of each camera. Once these are

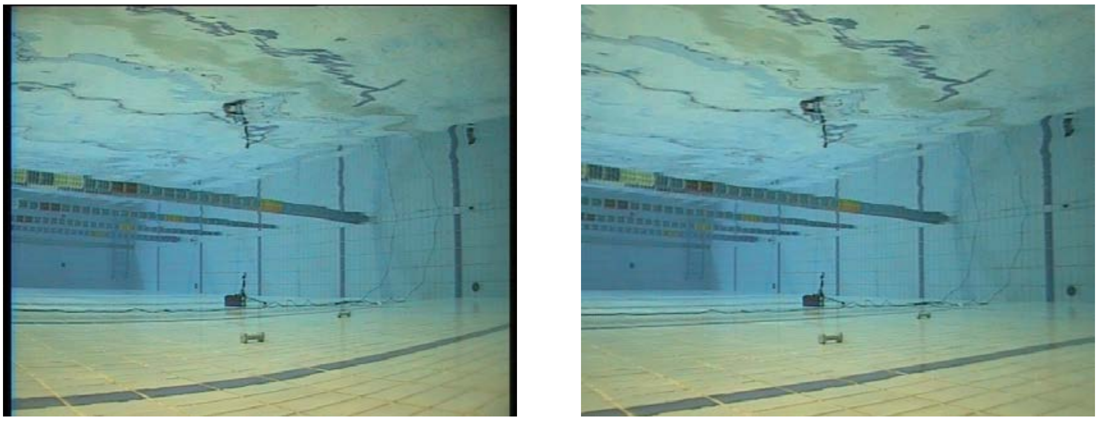


Figure 4.3: Left, a frame from a video recorded in an underwater environment. Right, the same frame after undistortion.

known, the rigid transformation relating the camera's and global coordinate systems, which generally do not coincide, may also be computed. A 3D point $P_G = (X, Y, Z)$ in the world reference frame will have the following coordinates in the camera's:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \tag{4.3}$$

or more compactly:

$$P = [R \,|\, \mathbf{t}] P_G, \tag{4.4}$$

where the joint rotation-translation matrix $[R \,|\, \mathbf{t}] \in R^{3\times 4}$ encodes the camera's extrinsic parameters. Furthermore, computational tools[3] may be employed to correct the distortion artefact in all frames of the captured video sequences, as exemplified in Figure 4.3. After performing undistortion, equations (4.1) and (4.3) may be combined, and expressed in a linear fashion as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \simeq \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{4.5}$$

or equivalently:

$$\mathbf{m} \simeq K[R \,|\, \mathbf{t}]\,\mathbf{M}\,, \tag{4.6}$$

[3] We employed the `undistort` function of the OpenCV Python library [60] for this task.

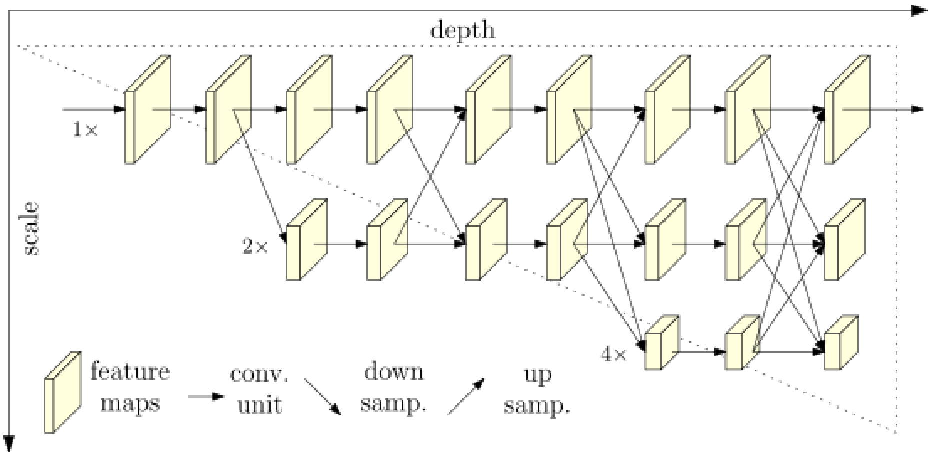


Figure 4.4: Architecture of High-Resolution Net (image from [19]).

where the symbol $\simeq$ means "equal up to a scale factor" and the vectors $\mathbf{m} = (u, v, 1)^\top$ and $\mathbf{M} = (X, Y, Z, 1)^\top$ are referred to as the homogeneous coordinates of p and P_G. Finally, by multiplying the matrices of intrinsic and extrinsic parameters, we obtain for each camera a projection matrix $\mathbf{P} \in \mathbb{R}^{3\times 4}$ such that $\mathbf{m} = \mathbf{PM}$.

4.2 High-Resolution Net

The synchronized videos acquired during the first step of our pipeline are passed as input to a convolutional pose estimator, employed to predict within each frame the position of anatomical keypoints such as the shoulders and knees. For this purpose, we made use of a top-down model known as High-Resolution Net (HRNet), presented by Sun et al. in [19], whose architecture is shown in Figure 4.4. Among the several PE approaches found in the literature (an overview of which was given in Chapter 3), this CNN was chosen due to its state-of-the-art performance on both the COCO [17] and MPII [27] datasets, as well as the excellent PyTorch implementation provided by the authors at https://github.com/leoxiaobin/deep-high-resolution-net.pytorch.

Detailed descriptions of High-Resolution Net (particularly the ways in which it differs from other existing PE models) and its use within our pipeline are provided in the following sections.

4.2.1 Comparison between HRNet and Previous Models

The most distinguishing architectural characteristic of High-Resolution Net concerns the way in which features are extracted from the input images, and combined during a forward pass to generate prediction heatmaps.

In other contemporary pose estimation approaches, this operation is commonly carried out by means of a two-step procedure. Firstly, input images are run through a series of subnetworks, each of which uses banks of convolutional filters to capture image features at a specific scale. Adjacent subnetworks are linked by a down-sampling layer, which halves the input's resolution before the next stage. The purpose of this "high-to-low resolution" process is to compute progressively higher-level features, enabling the network, through the decrease in resolution, to capture large-scale characteristics of the image, such as colors and shapes. Using the same notation as [19], let $\mathcal{N}_{sr}$ denote the subnetwork of the s^{th} stage, whose resolution is $\frac{1}{2^{r-1}}$ with respect to that of the input. A high-to-low resolution network consisting of four stages may then be represented as:

$$\mathcal{N}_{11} \ \rightarrow \mathcal{N}_{22} \ \rightarrow \mathcal{N}_{33} \ \rightarrow \mathcal{N}_{44} \tag{4.7}$$

After the last high-to-low stage, a high-resolution representation is recovered from the low-resolution ones through a "low-to-high resolution" process. This procedure is aimed at computing local, low-level features such as corners and lines, and its computational complexity may vary greatly depending on the model. For example, while in [36] the low-to-high process mirrors the high-to-low one in the number and shape of the convolutional filters used, in [21] it consists only of a few upsampling operations. Furthermore, a number of high-to-low, low-to-high network modules may be concatenated in order to increase the precision of final predictions.

HRNet differs from the approaches described above in that high-resolution representations are maintained throughout the entire network, rather than undergoing a downsampling process and being retrieved at a later stage. The network initially consists of a single high-resolution module; with each stage, one high-to-low branch is connected in parallel (rather than in series) to the main body. A 4-stage example of the resulting network structure, provided by the authors of [19], is shown below:

$$\begin{tikzcd}
\mathcal{N}_{11} \arrow[r] \arrow[dr] & \mathcal{N}_{21} \arrow[r] \arrow[dr] & \mathcal{N}_{31} \arrow[r] \arrow[dr] & \mathcal{N}_{41} \\
& \mathcal{N}_{22} \arrow[r] & \mathcal{N}_{32} \arrow[r] & \mathcal{N}_{42} \\
& & \mathcal{N}_{33} \arrow[r] & \mathcal{N}_{43} \\
& & & \mathcal{N}_{44}
\end{tikzcd} \tag{4.8}$$

As can be seen from (4.8), an additional, lower resolution is introduced within each stage. This approach means that different resolutions are handled by separate (albeit connected, see Section 4.2.2) parallel branches, rather than one being transformed into the other through scaling operations. This removes the need for a separate high-to-low process (as high resolution representations are preserved during a forward pass), leading to more precise spatial localization.

4.2.2 Multi-Scale Fusion

The use of multiple separate branches, each dedicated to a specific resolution, prevents possible imprecisions caused by rescaling operations. However, in order for the network's predictions to be as accurate as possible, both low and high-level representations should be considered, meaning features from across all scales should be taken into account. In HRNet, this is addressed by repeatedly sharing information across parallel high-to-low resolution subnetworks. This process, referred to as multi-scale fusion, is carried out in dedicated layers called exchange units.

Below is an example provided by the authors of [19]. The third network stage (which contains three parallel branches) was divided into three blocks, with an exchange unit placed after each block. In the following, let $\mathcal{C}_{sr}^{b}$ denote the convolutional filter of the r^{th} resolution, found in the b^{th} block of s^{th} stage, and $\mathcal{E}_{s}^{b}$ denote the corresponding exchange unit. The resulting architecture is given in the diagram below, while the structure of an exchange unit is shown in Figure 4.5.

$$\begin{tikzcd}
\mathcal{C}_{31}^{1} \arrow[dr] & & \mathcal{C}_{31}^{2} \arrow[dr] & & \mathcal{C}_{31}^{3} \arrow[dr] & \\
\mathcal{C}_{32}^{1} \arrow[r] & \mathcal{E}_{3}^{1} \arrow[ur] \arrow[r] \arrow[dr] & \mathcal{C}_{32}^{2} \arrow[r] & \mathcal{E}_{3}^{2} \arrow[ur] \arrow[r] \arrow[dr] & \mathcal{C}_{32}^{3} \arrow[r] & \mathcal{E}_{3}^{3} \\
\mathcal{C}_{33}^{1} \arrow[ur] & & \mathcal{C}_{33}^{2} \arrow[ur] & & \mathcal{C}_{33}^{3} \arrow[ur] &
\end{tikzcd} \tag{4.9}$$

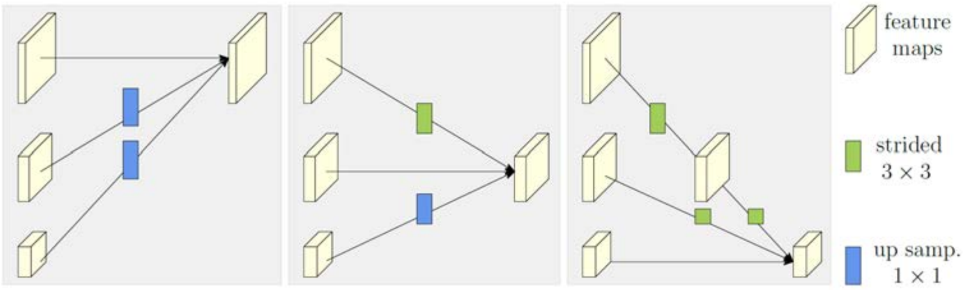


Figure 4.5: Structure of an HRNet exchange unit (image from [19]).

Each exchange unit takes as input a set of n (where n is the number of parallel branches in the stage) feature maps $\{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$ and returns another set $\{\mathcal{Y}_1, \ldots, \mathcal{Y}_n\}$, such that for each $k \in \{1, \ldots, n\}$ $\mathcal{Y}_k$ has the same resolution and shape as $\mathcal{X}_k$. The output representations are computed by combining information from all the inputs, according to the formula $\mathcal{Y}_k = \sum_{i=1}^{n} a(X_i, k)$. Depending on the values of i and k, function a performs either upsampling or downsampling operations on the input representations, so that their resolution matches the output's:

- If $i > k$, $\mathcal{X}_i$'s resolution is reduced using $(i - k)$ $3{\times}3$ convolutional filters with a stride of 2 (each of which halves the input's resolution).
- If $i < k$, $\mathcal{X}_i$ is first passed through a $1{\times}1$ convolutional filter in order to match the output number of channels. Its resolution is then increased via nearest neighbour upsampling.
- If $i = k$, a is the identity mapping.

The combined use of parallel branches and multi-scale fusion enables HRNet to preserve high-resolution feature maps, while still taking both high and low level information into account, leading to a more accurate estimation of joint locations.

4.2.3 Implementation Details

In testing their CNN, the authors of [19] developed several implementations of High-Resolution Net, all sharing a common baseline structure (shown in Figure 4.4) consisting of four stages with four parallel branches. Although versions of the network were trained on both the COCO and MPII datasets, only the former was considered in this work, as we found models based on the latter would often fail to produce accurate pose estimates in the presence of motion blur (a common artefact in the videos we analyze). The two available COCO-trained architectures, referred to as HRNet-W32 and HRNet-W48, differ solely in their size, with the second employing branches of significantly higher width (48, 96, 192 and 384 against 32, 64, 128 and 256) in last three stages[4]. As the use of the larger model provided no noticeable performance gain, we chose to make use of the computationally lighter W32 variant in our tests.

The network takes as input images of size 384 × 288 (4:3 aspect ratio), obtained by cropping the views of interest (in our case, the frames of the recorded walking trials) around a single person. During training, these bounding boxes are provided as annotation to the images in the COCO dataset, while at testing time they are generated by a person detector, as detailed in Section 4.2.4. For a given input, the CNN returns a set of 17 heatmaps (one for each keypoint in the COCO model) of size 96 × 72, whose intensity values, ranging from 0 to 1, express the per-pixel likelihood of the corresponding joint being found in each location of the output space. In order to improve prediction stability, we additionally evaluate a horizontally flipped version of each image, averaging the resulting heatmaps with the ones obtained for the original. This technique, referred to as flip test, is widely employed in recent pose estimation literature [21, 22, 36], and is credited in [36] with a 1% localization accuracy improvement on the MPII dataset. In accordance with specifications provided in [19], the final predictions for the 2D position of each joint were obtained by applying a shift of one quarter of a pixel to the coordinates of the associated heatmap's highest intensity response, in the direction of the second highest, and then multiplying them by 4 to recover the original input dimensions.

4.2.4 Person Detection

As mentioned previously, pose estimation in High-Resolution Net is carried out according to the top-down paradigm, meaning model inputs are assumed to be views of a single person. This makes it necessary, in order for the CNN to work correctly with generic images, to employ a person detection (PD) preprocessing step, drawing around each individual in the scene a bounding box on which the network can be run.

[4] The residual modules of the first stage have a width of 64 in both models.

Even on sequences such as the considered walking trials, in which only one subject is present in the foreground at any given time, this operation is required to obtain network inputs of the correct position and size.

Human detection is carried out in this work by means of a dedicated convolutional neural network, to which HRNet was connected in series. Specifically, we make use of a multi-class object detector (OD), capable of identifying and localizing instances of several common object categories (including people) in images and videos. A variety of pre-trained architectures of this kind are provided by the TensorFlow Object Detection API [61], an open-source framework built on top of the TensorFlow deep learning library [62]. Among these, we chose to employ a network[5] relying on a Faster R-CNN [63] backbone, providing, partially due to the use of efficiency-enhancing Inception v2 modules [64], an excellent compromise between accuracy and speed. Object detection is performed in the selected model according to the following two-step pipeline:

1. through the use of a Region Proposal Network (RPN), extrapolate from the input image a number of saliency areas in which an object is believed to reside;
2. identify the objects present in the scene by independently applying an image classifier to each formulated region proposal.

For a given input, the OD network returns a set of rectangular bounding boxes (identified by the coordinates of their upper-left and bottom-right corners), each of which is associated with a class label and a confidence score $c \in [0, 1]$, expressing the likelihood of the classification being correct. Only the person class is considered in this work; furthermore, in order to avoid spurious detections, we disregard all predictions for which c is lower than 0.5. Among the remaining boxes, the one associated with the highest score is assumed to contain the individual of interest. Example outcomes of this procedure are showcased, for both a laboratory and an underwater environment, in Figure 4.6.

As the evaluated region proposals do not in general have the required 384×288 shape, an additional processing step is required before pose estimation can be performed on the outputs of the PD network. In [19], this operation takes the form of an

[5] Referred to as `faster_rcnn_inception_v2_coco` in the TensorFlow Object DetectionZoo.

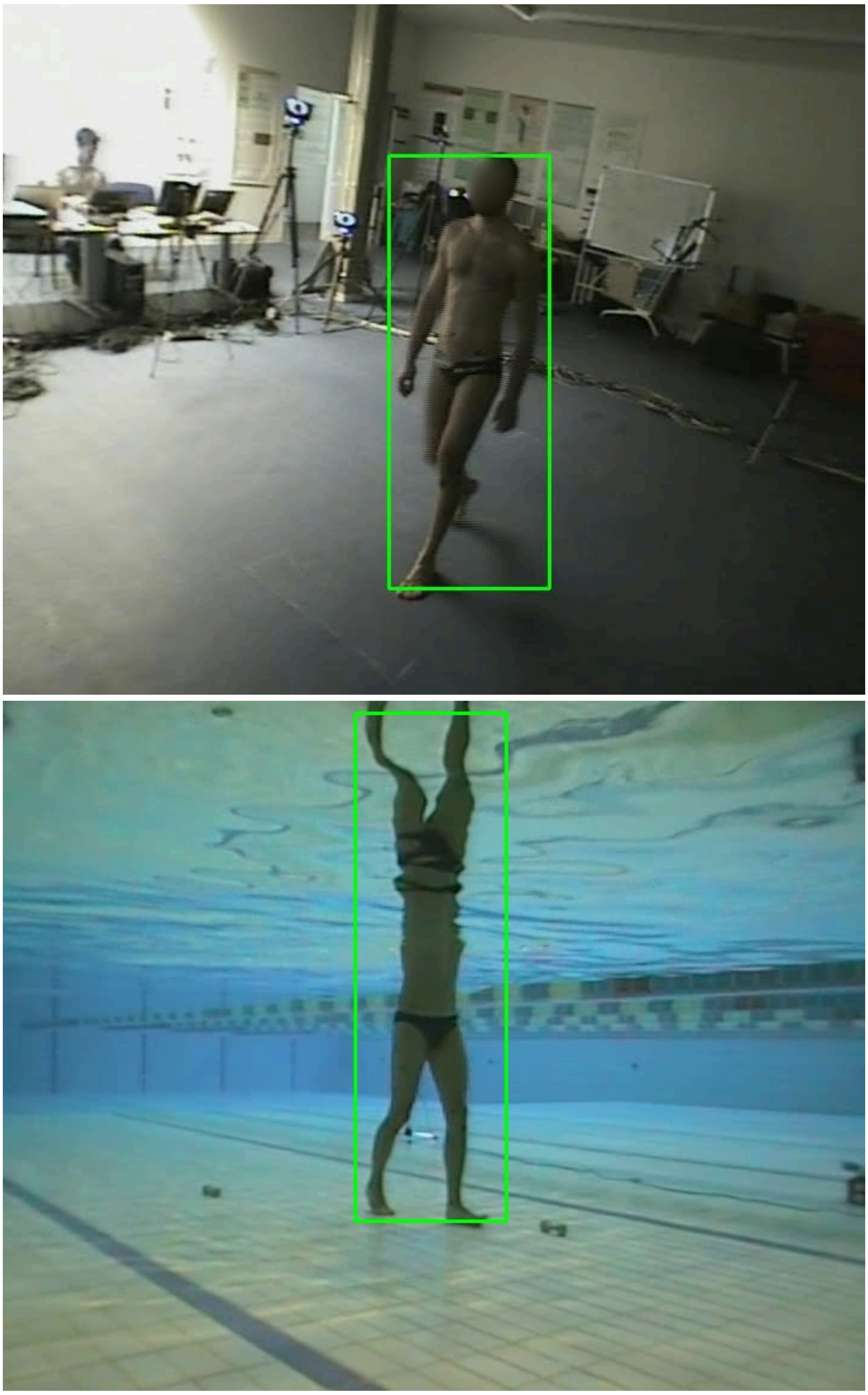


Figure 4.6: Example of bounding boxes generated by the person detector from views of a walking subject, in both a laboratory (top) and underwater (bottom) environment.

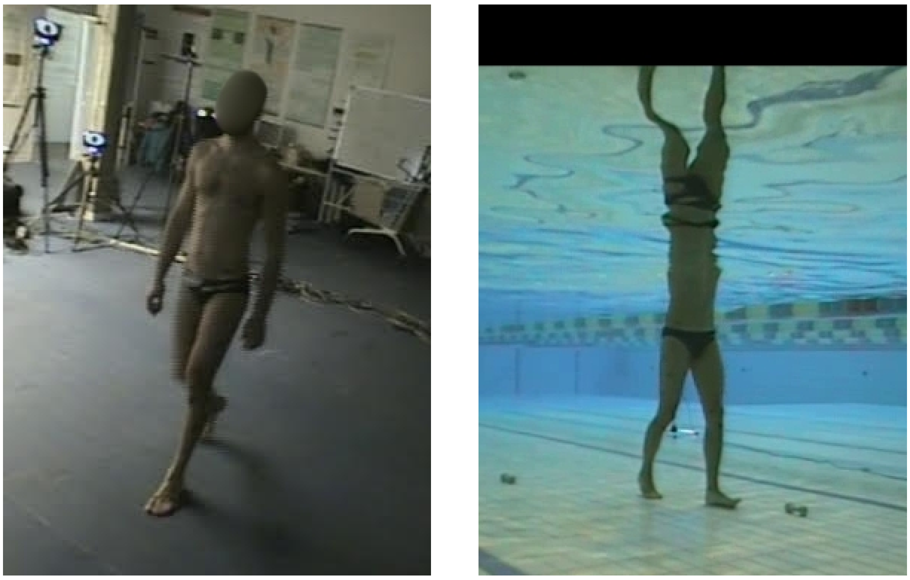


Figure 4.7: Properly sized HRNet inputs, obtained from the bounding boxes shown in Figure 4.6 by means of an affine transform. The black bar in the UW image (right) was added to achieve the desired 4:3 aspect ratio despite the frame's upper border being reached.

affine transform which, taking as input the center and shape of the considered bounding box (where the latter is defined as the box's height normalized by 200 pixels), returns a correctly sized view of the scene in which the subject occupies a prominent position, as shown in Figure 4.7.

A simple linear interpolation technique was employed to obtain valid HRNet inputs for frames on which the OD network was unable to produce person bounding boxes with a sufficiently high confidence. Consider a video sequence $V = \{1, \ldots, n\}$, and its subset S on whose elements detection was successful. Given a frame $i \notin S$ on which PD failed, let:

$$p = \max\{j \in S \mid j < i\}, \quad s = \min\{j \in S \mid j > i\}, \tag{4.10}$$

represent the closest successful predecessor and successor of i, having center and scale $(c_p,\, s_p)$ and $(c_s,\, s_s)$. The parameters $(c_i,\, s_i)$ characterizing the missing bounding box were estimated as:

$$\begin{cases} c_i = w_p c_p + w_s c_s \\ s_i = w_p s_p + w_s s_s \,, \end{cases} \tag{4.11}$$

where:

$$\begin{cases} w_p = \dfrac{s-i}{s-p} \\ w_s = \dfrac{i-p}{s-p} \, . \end{cases} \tag{4.12}$$

In other words, the bounding box of frame i was computed as the weighted average (in both center and scale) of the boxes associated to p and s, where the weight assigned to each successful frame was set to be inversely proportional to its distance from i.

Frames lacking a successful predecessor were dealt with by setting $w_p = 0$ and $w_s = 1$. Analogously, in the absence of a valid successor we selected $w_p = 1$ and $w_s = 0$. These events, usually caused by a subject having yet to enter or already left the camera's field of view, were assumed to never occur concurrently, as the opposite would require the person detector to fail on all frames of the considered sequence.

Out-of-Water Person Detection

When applied to recordings of trials performed in an out-of-water (OW) environment, the person detector was in most cases able to correctly locate the walking subject. Although failed detections did occasionally occur (mainly due to the the substantial amounts of noise and motion blur affecting the considered OW sequences), in these instances the employed interpolation technique proved capable of generating serviceable bounding boxes.

It is worth noting that, on videos taken from certain viewpoints, the PD network registered a successful detection even for frames where the subject was not visible, due to the presence of a human operator in the background (see Figure 4.6 (top) for reference). While this behaviour had no influence on the pose estimator's performance, it did prevent us from employing the detector's output to automatically pinpoint the frames in which, for a given trial, the subject entered and exited the scene (an operation which had to be carried out manually instead).

Underwater Person Detection

Due to the way in which underwater (UW) gait trials were carried out, person detection proved significantly more difficult in this setting than in a controlled laboratory environment. As can be seen from Figure 4.11, subjects were only submerged up their chest, meaning their head, shoulders and arms were out of view of the cameras in most cases. Furthermore, the reflective properties of the water surface often caused their legs to be vertically mirrored in the frame's upper half, introducing a foreign element in the scene that the network (developed for use in OW conditions) was not trained to recognize.

Despite these difficulties, we found that a subject's lower body being visible was generally sufficient for the person detector to correctly locate them, allowing us, as discussed in the following, to successfully apply our pipeline to underwater scenes. While the occlusion of the upper limbs resulted in a lower average classification score, and consequently a greater rate of failed predictions in comparison to an OW setting, suitable HRNet inputs could still be obtained by means of box interpolation. Exceptions to this were rare instances in which undesired scene elements, such as a fully submerged swimmer in the background, caused the network to entirely miss the subject of interest, while still indicating a successful detection.

Lastly, it should be noted that even on frames in which the subject was correctly located, the aforementioned reflection artefact was frequently (when present) recognized as part of the detected individual, resulting, as exemplified in Figure 4.6, in unnaturally tall bounding boxes. This behaviour, owed to the PD network being unable to generalize to a UW environment, was however found to have little influence on our system's performance. As showcased in Section 4.2.5, pose estimation quality was only significantly degraded for upper-body limbs, which due to their limited visibility were not taken into consideration in this work. This conclusion is particularly convenient when considering that, because of the task's high specificity, PD approaches developed especially for underwater use are currently missing in the literature, as it allows more common OW techniques to be effectively repurposed for this goal.

4.2.5 Network Output

As discussed in Chapter 3, the heatmap output representation employed by HRNet allows the CNN to convey a measure of confidence on each predicted keypoint position. This feature is particularly useful when dealing with self-occlusions, the phenomenon by which a keypoint is blocked from the view of a camera by another body part.

To better understand this, consider first the ideal situation of Figure 4.8, in which the network input (shown on the left-hand side) consists of a frontal view of a subject

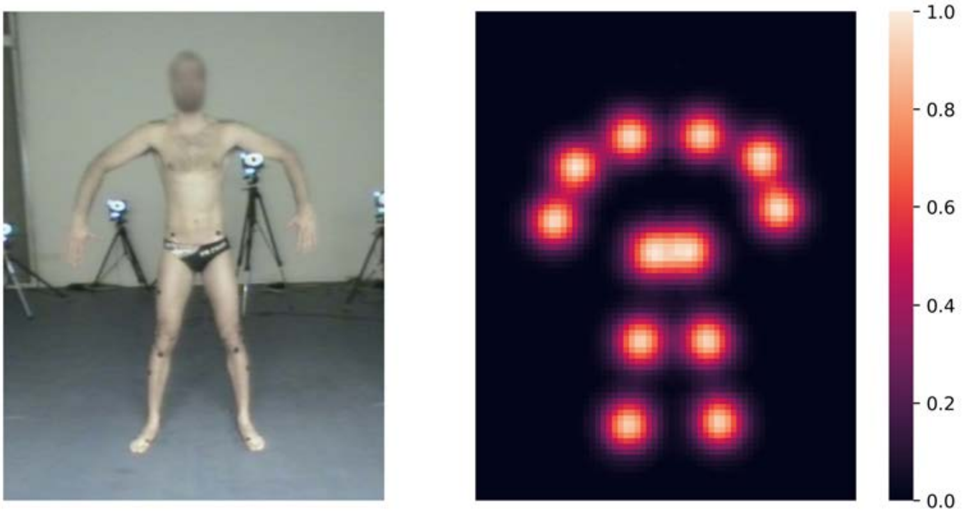


Figure 4.8: Left, a subject in reference pose in an out of water environment. Right, result of performing 2D pose estimation on the left image.

in reference pose. Since all the body joints defined in the COCO model (shoulders, elbows, wrists, hips, knees and ankles) are clearly visible, the corresponding heatmaps (which for visualization convenience have been superimposed on the right-hand side) have a well-defined shape, as well as an intensity peak approaching the maximum value of 1.

Figure 4.9 (left) displays a more realistic input configuration where, due to the camera placement and the subject's walking movement, the right elbow and wrist are occluded from view. As can be seen from the right-hand image[6], non-visible keypoints are associated with irregularly shaped heatmaps, covering a wider area with significantly lower intensity values. This may be interpreted as the network expressing uncertainty over the joints' location, assigning a similar likelihood to several output locations.

The examples discussed above showcase how, thanks to the use of heatmaps, CNN-based pose estimators may implicitly distinguish between visible and occluded keypoints, automatically assigning high confidence scores to the former and low ones to the latter. As described in Section 4.3, this fact is exploited by our system when triangulating the 3D position of each joint, in order to take high-quality 2D predictions into greater consideration.

[6] For clarity, only the heatmaps corresponding to the right elbow and wrist are shown.

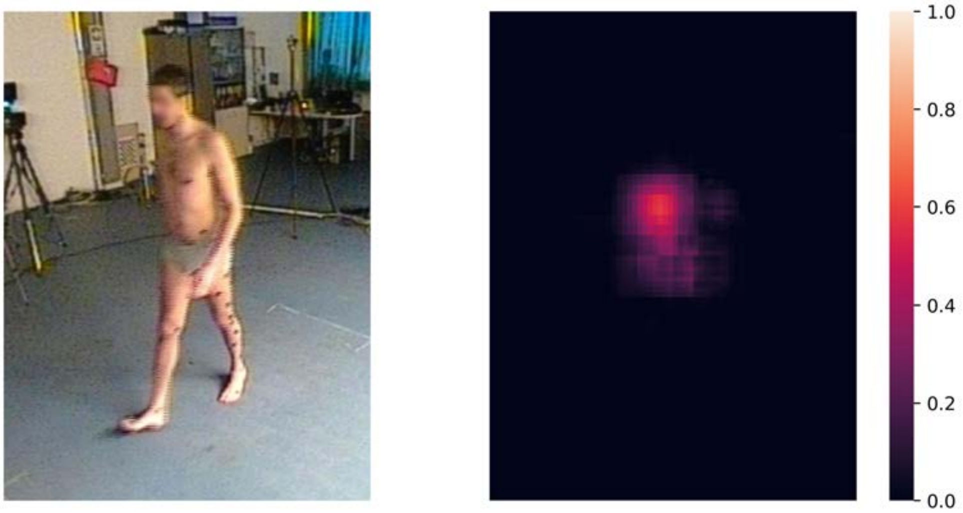


Figure 4.9: Left, a view of a walking subject in an OW environment, in which the left elbow and wrist are occluded from view. Right, the irregularly shaped heatmaps generated by HRNet for the aforementioned non-visible keypoints.

Out-of-Water Pose Estimation

Figure 4.10 shows two 2D pose estimates generated by HRNet from views of walking subjects in an out-of-water (OW) environment. Keypoints were numbered according to the COCO specification (see Figure 4.13 for reference), with joints 0 through 4 (referring to facial features) omitted for clarity. Each prediction was assigned a confidence score c, defined as the peak intensity value of the corresponding heatmap, and represented by a round marker displayed as:

- red if $c < 0.25$;
- orange if $c \in [0.25,\ 0.5)$;
- yellow if $c \in [0.5,\ 0.75)$;
- green if $c \geq 0.75$.

Despite the substantial amounts of noise and motion blur affecting the frames' quality, visible keypoints were consistently placed in anatomically correct positions (as well as associated to high heatmap responses) by HRNet. Furthermore, it can be seen that even in the presence of occluded joints (such as the left knee in the left-hand image, or the right ankle in the right-hand one) the network was often

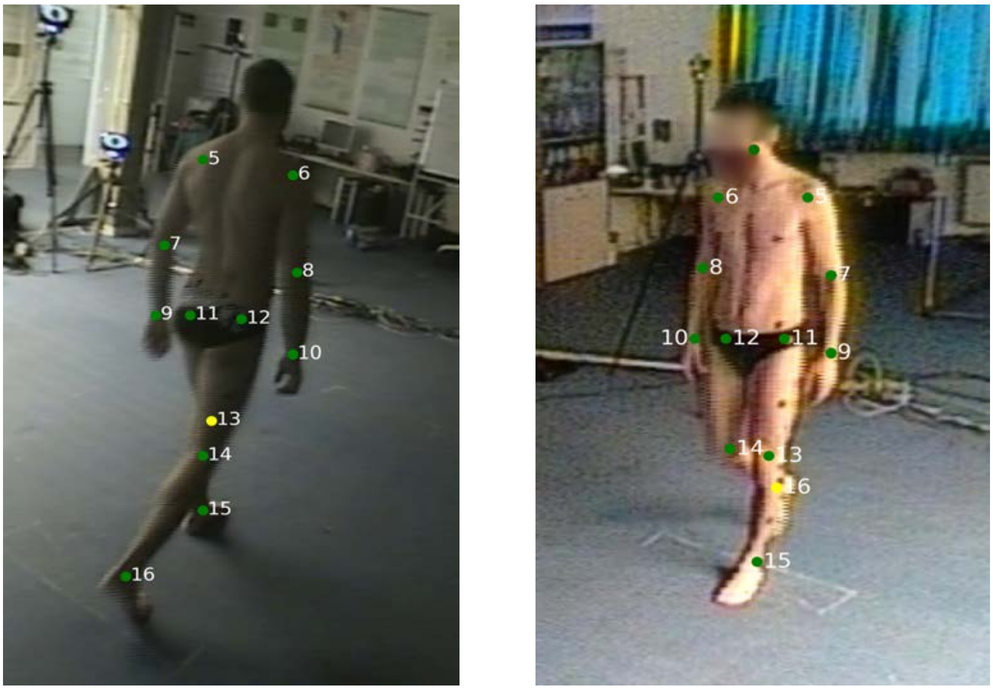


Figure 4.10: 2D pose estimates generated by HRNet from two views of walking subjects in an OW environment.

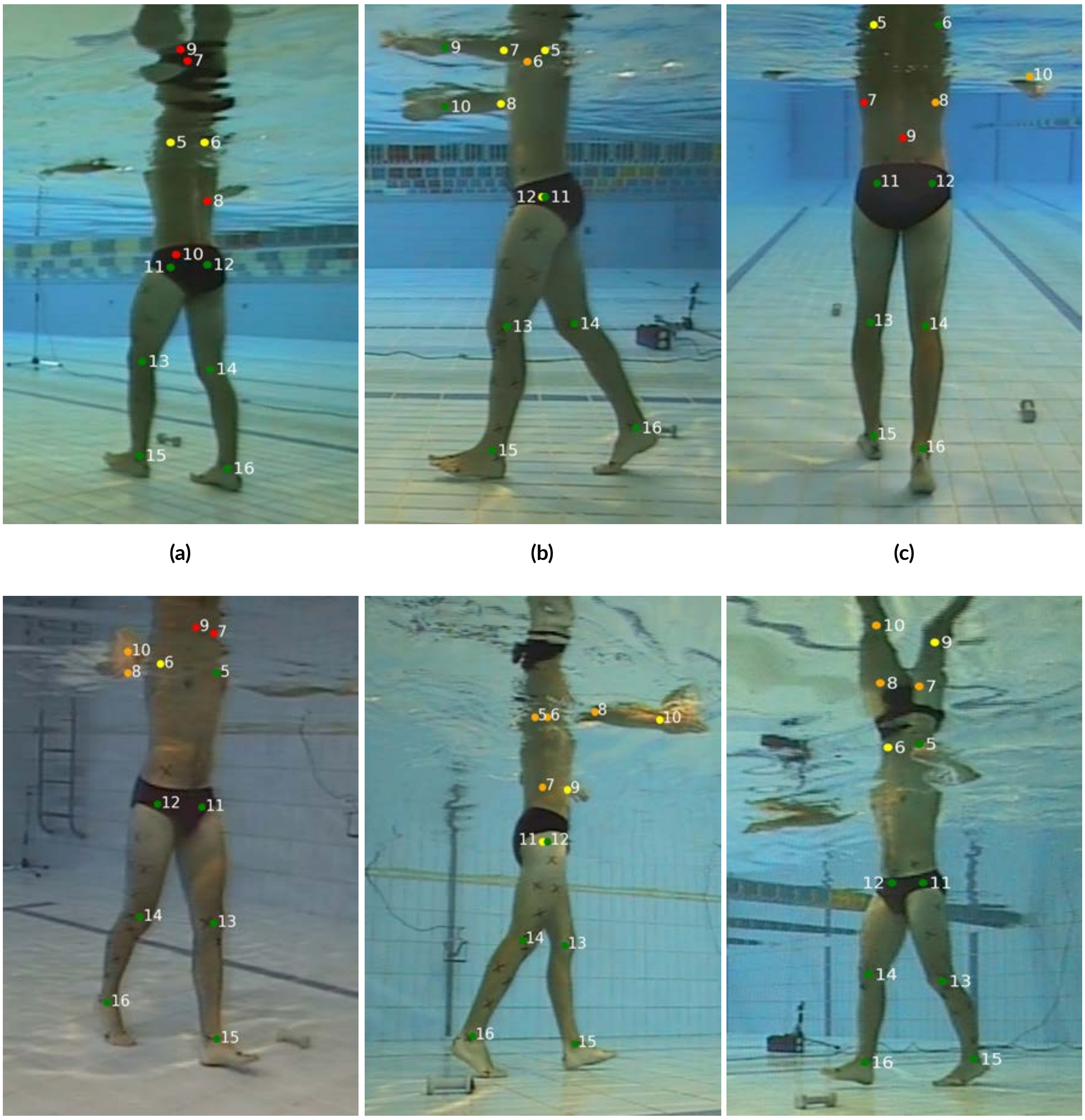
able to formulate relatively accurate predictions, albeit with lower confidence. This is made possible by the intrinsic ability of CNNs to understand images on a global level, inferring the location of non-visible keypoints from contextual cues such as the subject's stance and the configuration of other body parts.

Underwater Pose Estimation

The task of full-body pose estimation becomes considerably more challenging when performed in an underwater (UW) environment. In order to highlight the difficulties faced by our system in this setting, we provide in Figure 4.11 a set of 2D pose estimates, derived by HRNet from six differently oriented views of a walking subject[7].

The most striking difference from the OW results shown in Figure 4.10 is the much poorer performance of the CNN on joints belonging to the upper body. Because in the considered gait trials subjects were submerged up to chest level, their arms are often non-visible throughout the recorded sequences. This issue, exacerbated by the constant movement of the water, resulted in the imprecise placement of the elbow

[7] The same numbering and colouring conventions as Figure 4.10 were employed.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.11: 2D pose estimates generated by HRNet from six views of a walking subject in an underwater environment.

and wrist joints across all cameras, with somewhat accurate predictions only being made, with fairly low confidence, in (b) and (e).

Another critical, UW-specific complication is the occurrence of total internal reflection, a phenomenon causing the water surface to exhibit reflective properties from certain viewpoints, resulting in the appearance of a vertically mirrored image of the subject's lower body. The consequences of this are especially noticeable in (a) and (f), where keypoints associated with the (non-visible) upper limbs were erroneously positioned within the reflection artefact (in particular, note how in (f) the elbow and wrist joints were placed in correspondence of the reflected hips and knees).

Due to the issues discussed above, we decided in this work to discard upper-body predictions made in a UW setting. Fortunately, we found that even in the presence of water HRNet was able to consistently produce high-quality estimates for the hips, knees and ankles, allowing for the accurate triangulation of lower body keypoints.

4.3 Triangulation

In order to meaningfully quantify the subjects' movement throughout the recorded sequences, it is necessary to extrapolate their 3D configuration from the 2D pose estimates formulated by High-Resolution Net. In this work, this operation is carried out by means of a triangulation routine which, taking as input the predictions made for a keypoint's location across n synchronized views (where n is the number of cameras employed in the acquisition step) returns an estimate of its 3D position in the calibrated space.

The efficient *linear-eigen* method [65], outlined in Section 4.3.1, was chosen as the basis of the triangulation process. Leveraging positional data gathered during the cameras' calibration, this approach provides a closed-form solution for the 3D location of each joint, allowing the subject's pose to be recovered with minimal computational complexity. In its original formulation however, this algorithm assigns equal importance to each considered 2D point. As discussed at the beginning of this chapter, we instead wish to take higher-quality predictions into greater consideration, exploiting the confidence information provided by HRNet with each evaluation. For this purpose, we make use of a simple weighted variant of linear-eigen, a description of which is given in Section 4.3.2.

4.3.1 Linear-Eigen Triangulation

Consider the example (only displaying two cameras) provided in Figure 4.12. Given n sets of pixel coordinates $m_i = [u_i, v_i]$, $i \in \{1, \dots, n\}$, as well as the camera matrices

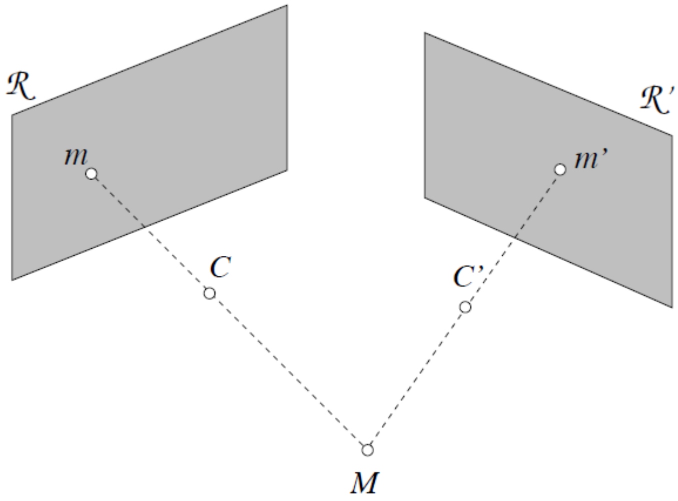


Figure 4.12: Example of 3D triangulation from two camera views (image from [66]).

$\{\mathbf{P}_i\}_{i=1}^{n}$, we are interested in determining a 3D point $M = [x, y, z]$, of which all the m_is are assumed to be the projection. For each $(m, \mathbf{P})$ pair we have from (4.6):

$$\mathbf{m} \simeq \mathbf{P}\mathbf{M}, \tag{4.13}$$

where $\mathbf{m} = [u, v, 1]$ and $\mathbf{M} = [x, y, z, 1]$ are referred to as the homogeneous coordinates[8] of m and M respectively. Letting $\mathbf{p}_j^\top$ denote the j^{th} row of $\mathbf{P}$, (4.13) can be reformulated as:

$$\mathbf{m} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \simeq \begin{bmatrix} \mathbf{p}_1^\top \\ \mathbf{p}_2^\top \\ \mathbf{p}_3^\top \end{bmatrix} \mathbf{M} = \begin{bmatrix} \mathbf{p}_1^\top \mathbf{M} \\ \mathbf{p}_2^\top \mathbf{M} \\ \mathbf{p}_3^\top \mathbf{M} \end{bmatrix}. \tag{4.14}$$

In Cartesian coordinates, this is equivalent to:

$$\begin{cases} u = \dfrac{\mathbf{p}_1^\top \mathbf{M}}{\mathbf{p}_3^\top \mathbf{M}} \\ v = \dfrac{\mathbf{p}_2^\top \mathbf{M}}{\mathbf{p}_3^\top \mathbf{M}}, \end{cases} \tag{4.15}$$

[8] As detailed in [66], the use of homogeneous coordinates allows projective transformations to be expressed by a single matrix multiplication.

which yields, after a few algebraic steps:

$$\begin{cases} (\mathbf{p}_1 - u\mathbf{p}_3)^\top \mathbf{M} = 0 \\ (\mathbf{p}_2 - v\mathbf{p}_3)^\top \mathbf{M} = 0. \end{cases} \tag{4.16}$$

From each 2D point it is therefore possible to derive two homogenous equations in the coordinates of $\mathbf{M}$. Repeating the procedure described above for each m_i, we obtain the following system of $2n$ equations in four unknowns:

$$\begin{cases} (\mathbf{p}_1^1 - u_1\mathbf{p}_3^1)^\top \mathbf{M} = 0 \\ (\mathbf{p}_2^1 - v_1\mathbf{p}_3^1)^\top \mathbf{M} = 0 \\ \quad\vdots \\ (\mathbf{p}_1^n - u_n\mathbf{p}_3^n)^\top \mathbf{M} = 0 \\ (\mathbf{p}_2^n - v_n\mathbf{p}_3^n)^\top \mathbf{M} = 0, \end{cases} \tag{4.17}$$

or alternatively in matrix form:

$$\mathbf{AM} = \mathbf{0}_{2n\times 1}, \quad \mathbf{A} = \begin{bmatrix} (\mathbf{p}_1^1 - u_1\mathbf{p}_3^1)^\top \\ (\mathbf{p}_2^1 - v_1\mathbf{p}_3^1)^\top \\ \vdots \\ (\mathbf{p}_1^n - u_n\mathbf{p}_3^n)^\top \\ (\mathbf{p}_2^n - v_n\mathbf{p}_3^n)^\top \end{bmatrix} \in \mathbb{R}^{2n\times 4}. \tag{4.18}$$

System (4.17) is overdetermined[9] for any $n > 2$ and, due to the presence of noise, cannot in general be solved exactly. Instead, a solution is determined in a least-squares sense, finding a vector $\mathbf{M}$ such that:

$$\mathbf{M} \in \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^{4\times 1}} \|\mathbf{Ax}\|^2, \text{ subject to } \|\mathbf{x}\| = 1. \tag{4.19}$$

The additional constraint on $\|\mathbf{x}\|$ was placed in order to avoid the the trivial result $\mathbf{M} = \mathbf{0}$. The solution to this problem can be shown to be the last column of matrix $\mathbf{V}$, where $\mathbf{A} = \mathbf{UDV}^\top$ is the Singular Value Decomposition (SVD, [67]) of $\mathbf{A}$, or equivalently the eigenvector of $\mathbf{A}^\top\mathbf{A}$ associated to its smallest eigenvalue (hence the

[9] Has more equations than unknowns.

algorithm's name). A detailed proof can be found in [68, Appendix 5].

4.3.2 Weighted Triangulation

As discussed in Section 4.2.3, each joint prediction made by High-Resolution Net is associated with a 2D heatmap, measuring the per-pixel likelihood of the joint being found in each output location. In triangulating the 3D position of a keypoint, we wish to take this information into account, giving a lower importance to low-confidence predictions, such as the ones obtained in the presence of occlusions.

For a given joint and time instant, let w_i denote the maximum value of the heatmap produced by HRNet for the i^{th} synchronized view[10]. We weight each camera's contribution in (4.17) according to the w_is, obtaining the system:

$$\begin{cases} w_1(\mathbf{p}_1^1 - u_1\mathbf{p}_3^1)^\top \mathbf{M} = 0 \\ w_1(\mathbf{p}_2^1 - v_1\mathbf{p}_3^1)^\top \mathbf{M} = 0 \\ \quad\vdots \\ w_n(\mathbf{p}_1^n - u_n\mathbf{p}_3^n)^\top \mathbf{M} = 0 \\ w_n(\mathbf{p}_2^n - v_n\mathbf{p}_3^n)^\top \mathbf{M} = 0. \end{cases} \tag{4.20}$$

Note that for each $i \in \{1, \ldots, n\}$, equations $2i$ and $2i+1$ are derived from the same camera, and are thus assigned the same weight. A least-squares solution to (4.20) can be found by determining the vector $\mathbf{M}$ such that:

$$\mathbf{M} \in \underset{\mathbf{x}\in\mathbb{R}^{4\times 1}}{\arg\min} \sum_{i=1}^{n} w_i(\mathbf{A}_i^\top \mathbf{x})^2, \quad \text{subject to } \|\mathbf{x}\| = 1, \tag{4.21}$$

or equivalently:

$$\mathbf{M} \in \underset{\mathbf{x}\in\mathbb{R}^{4\times 1}}{\arg\min} \left\| \mathbf{W}^{\frac{1}{2}} \mathbf{A}\mathbf{x} \right\|^2, \ \text{subject to } \|\mathbf{x}\| = 1, \tag{4.22}$$

[10] Although in this work the weight associated with each camera was defined as the maximum intensity response of the corresponding heatmap, other choices are possible. For instance, a logarithmic scaling of the w_is could be employed to more aggressively penalize low-confidence predictions.

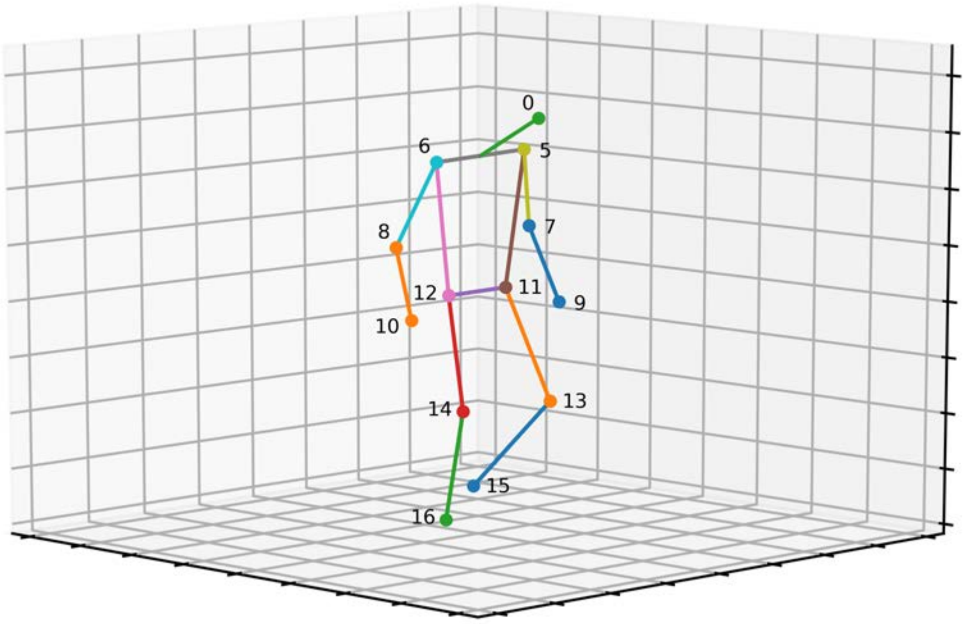


Figure 4.13: Example of a 3D pose estimate, based on the COCO model, obtained by weighted linear-eigen triangulation from six synchronized views of a subject walking in an OW environment.

where $\mathbf{W}$ is the diagonal weight matrix:

$$\mathbf{W} = \operatorname{diag}(w_1, w_1, w_2, w_2, \ldots, w_n, w_n). \tag{4.23}$$

Thus, weighted linear-eigen triangulation can be carried out using the same pipeline described in Section 4.3.1, performing SVD on $\mathbf{B} = \mathbf{W}^{\frac{1}{2}}\mathbf{A}$ rather than $\mathbf{A}$. Futhermore, because (4.22) has a closed-form solution, a suitable $\mathbf{M}$ can be determined very efficiently using analytical solvers [69].

Out of Water Environment

Figure 4.13 shows an example 3D pose estimate, obtained by means of weighted triangulation from six synchronized views of a subject walking in an out-of-water environment. The relations between keypoints and their numbering are based on the COCO model, which uses 17 joints arranged in a skeleton-like structure by means of pairwise connections[11]. Note that, while for clarity joint 0 (denoting the nose)

[11] Joints 1 through 4, referring to the eyes and ears, are not shown.

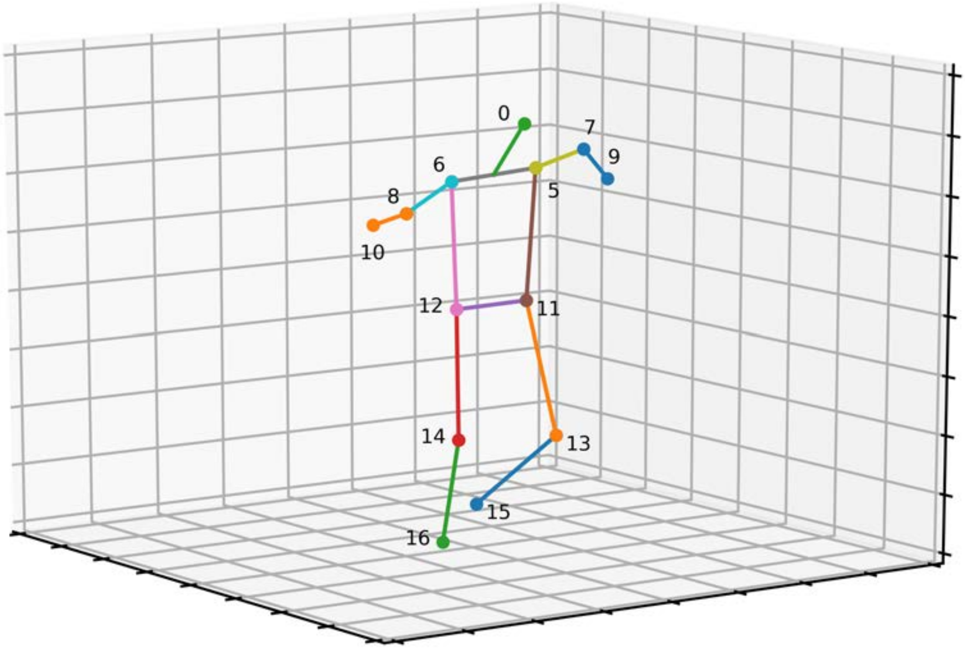


Figure 4.14: Example of a an unnatural 3D pose estimate, obtained by weighted linear-eigen triangulation from six synchronized views of a subject walking in a UW environment.

was linked to the midway point of the segment connecting the shoulders, the COCO model does not employ a neck keypoint (hence the absence of a numbered marker).

Underwater Environment

As discussed in Section 4.2.5, it is generally difficult in a UW environment to obtain meaningful 2D predictions for joints belonging to the shoulders and arms, due to reflections in the water surface as well as ripples induced by the subject's movement. Furthermore, because these body parts are often simultaneously hidden from the view of all cameras, no high-confidence estimate can be made to compensate for low-quality ones, resulting, as exemplified in Figure 4.14, in anatomically unnatural 3D upper body configurations. Importantly for the purposes of this work however, predictions made on lower-body keypoints are not affected by these problems, allowing for the accurate tracking of the abdomen and legs.

4.3.3 Considerations

The main disadvantage of the triangulation approach described in Section 4.3.2 is its potential inaccuracy, caused by the minimization of a purely algebraic objective. In the literature, it is common to employ a geometric loss function instead; for instance, Zhang's camera calibration procedure [70] iteratively optimizes the reprojection error:

$$\varepsilon(\mathbf{M}) = \sum_{i=1}^{N} \left\| \begin{bmatrix} u_i \\ v_i \end{bmatrix} - \begin{bmatrix} \frac{\mathbf{p}_1^{i\top}\mathbf{M}}{\mathbf{p}_3^{i\top}\mathbf{M}} \\ \frac{\mathbf{p}_2^{i\top}\mathbf{M}}{\mathbf{p}_3^{i\top}\mathbf{M}} \end{bmatrix} \right\|^2, \tag{4.24}$$

using the linear-eigen solution only as a starting point. Due to the reasons discussed next however, we decided against the adoption of this techinque in this thesis.

In [70], camera parameters are recovered by detecting a pattern of known dimensions in several positions and orientations. Pattern elements are designed to be easily recognizable (e.g. the corners of a chessboard) and are all simultaneously visible in each frame. By contrast, especially when dealing with multiple synchronized views, joints occlusions are a common occurrence in pose estimation, leading to low confidence predictions in which the location of a keypoint is known only approximately. The minimization of $\varepsilon(\mathbf{M})$ may therefore lead to the overfitting of imprecise predictions. Moreover, because our pipeline includes a separate precision-enhancing routine (see Section 4.4), the non-linear refinement employed in [70] would be redundant.

4.4 Prediction Refinement

Given a set of 2D joint predictions, obtained by High-Resolution Net from a number of synchronized camera views, the procedure described in the previous section yields 17 sets of 3D coordinates, providing an estimate of the subject's pose in each frame. It must be pointed out however that the COCO and MPII datasets, on which most contemporary pose estimators are trained, were not built for clinical applications, but rather for tasks such as activity recognition and pedestrian detection, which only require approximate joint positions. As a result, the triangulated HRNet predictions are generally not accurate enough for the clinically-oriented tracking of a human subject.

To alleviate this issue, we introduce an original prediction-refinement routine, which

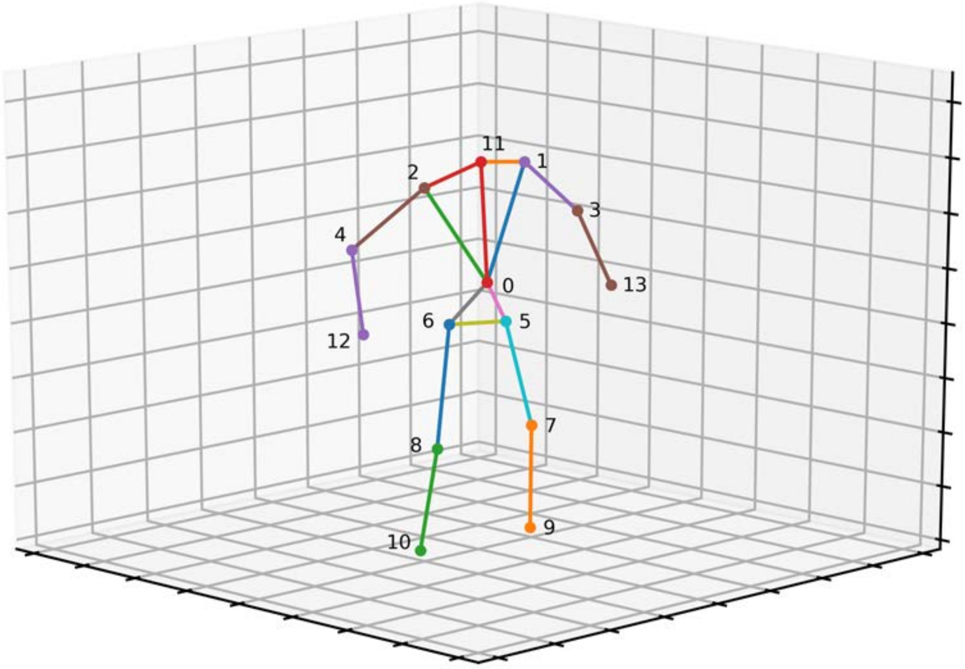


Figure 4.15: Example of a subject-specific articulated model. Each coloured segment denotes a rigid body part, whose length is not modified during the optimization process.

uses the computed 3D pose estimates to modify the configuration of a subject-specific articulated (SSA) model. This approach is qualitatively very similar to the matching procedure described in Section 2.3: the SSA representation (of which an example is provided in Figure 4.15) closely resembles the anatomical model employed in markerless motion capture and was, in fact, directly derived from it. Crucially however, while in the MMC pipeline a subject is described by both morphological and kinematic data, we discard the former in this work, retaining only information on the location of joint centers. This step, made necessary by the lack of volumetric information in the COCO model, also results in a substantially lower computational complexity.

As can be seen by comparing Figures 4.13 and 4.15, the COCO and SSA models exhibit significant differences. Most notably, the latter employs two additional keypoints, with the pelvis and neck joints (0 and 11, respectively) not being found in the former[12]. As a result, the hips, which in SSA are connected to the pelvis, are linked directly to the shoulders in COCO. In order to match differently structured models,

[12] Joints 0 through 4 in the COCO model, referring to the nose, eyes and ears, are disregarded in this thesis.

we define the bijection:

$$\begin{aligned} &f : D \to C \\ &D = \{1, 2, \ldots, 13\} \setminus \{11\}, \;\; C = \{5, 6, \ldots, 16\}, \end{aligned} \tag{4.25}$$

assigning to each SSA keypoint the index of its COCO counterpart; for example, we have for the left knee $f(7) = 13$. Lacking an equivalent in the COCO model, joints 0 and 11 are not included in the function domain.

The proposed matching algorithm was implemented in Python, making use of the SLSQP (Sequential Least SQuares Programming, [71]) solver. Taking as input:

- a subset $\{a_i\}_{i \in J}$ of joints from an articulated model $A = \{a_0, \ldots, a_{13}\}$, such that $J \subseteq D$;
- a 3D pose estimate $P = \{p_0, \ldots, p_{16}\}$;
- a set of SSA segments $S = \{(r_i,\, s_i)\}_{i=1}^{n}$, defined by the indexes of their endpoints, such that for all i:
  - $r_i,\, s_i \in \{0, 1, \ldots, 13\}$;
  - joints $r_i,\, s_i$ are connected in Figure 4.15.

the routine returns an articulated model $A' = \{a'_0, \ldots, a'_{13}\}$ such that:

$$\begin{aligned} &A' \in \underset{B=\{b_0,\ldots,b_{13}\}}{\arg\min} \sum_{i \in J} \|b_i - p_{f(i)}\|^2 \\ &\text{subject to: } \; \|b_{r_i} - b_{s_i}\| = \|a_{r_i} - a_{s_i}\| \;\; \forall\, (r_i, s_i) \in S. \end{aligned} \tag{4.26}$$

In other words, the algorithm determines the configuration of A that minimizes the sum of squared distances between the selected SSA joints and the corresponding COCO keypoints, while at the same time satisfying the articulated model's rigidity constraints by preserving the length of the chosen segments.

4.4.1 Full-Body Model Matching

In its first implementation, our prediction-refinement routine performed model optimization on a full-body scale, meaning the matching algorithm was applied to the entire subject-specific articulated model ($J = D$) and all SSA segments were included

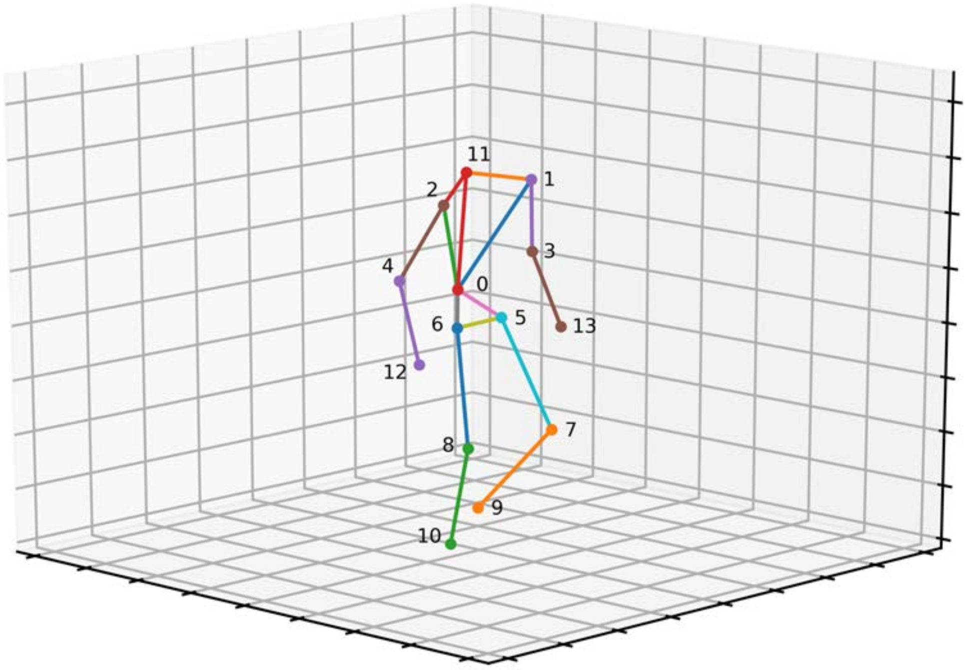


Figure 4.16: Example of an anatomically unnatural pose, obtained by matching the articulated model shown in Figure 4.15 to the pose estimate displayed in Figure 4.13 in a full-body manner.

in the hard constraint set S. However, we found that even in an out-of-water environment, in which the position of upper-body joints could be triangulated correctly, this approach often yielded anatomically incorrect configurations. This behaviour may be explained by considering that the 3D pose estimates generated from views of a subject are generally shorter in height and narrower at the shoulders than the corresponding articulated model[13]. Due to the constraints placed on segments of the abdomen and torso, it is therefore impossible for the algorithm to move the shoulder and hip SSA joints closer to their COCO equivalents without altering the proportions of the upper body. As a result, the pelvis and neck SSA keypoints, which as previously discussed do not contribute to the loss function (4.26), are pushed outwards, leading to unnaturally bent poses such as the one shown in Figure 4.16.

[13] We speculate this may be caused by a different definition of the shoulder joints in the two models.

4.4.2 Two-Step Model Matching

To address the shortcomings of full-body matching, we develop a modified prediction refinement procedure which, given an articulated model $A = \{a_0, \dots, a_{13}\}$ and a 3D pose estimate $P = \{p_0, \dots, p_{16}\}$, matches the former to the latter in two steps:

1. Apply the matching algorithm to the the lower body joints (pelvis, hips, knees and ankles), setting:
   - $J = \{0,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10\}$
   - $S = \{(0,5),\ (0,6),\ (5,6),\ (5,7),\ (6,8),\ (7,9),\ (8,10)\}$
2. Without modifying the position of lower-body joints, apply the matching algorithm to the upper-body keypoints (neck, shoulders, elbows and knees) of the articulated model returned by step 1, setting:
   - $J = \{1,\ 2,\ 3,\ 4,\ 11,\ 12,\ 13\}$
   - $S = \{(0,1),\ (0,2),\ (0,11),\ (1,11),\ (2,11),\ (1,3),\ (2,4),\ (3,13),\ (4,12)\}$

In order to provide the procedure with a starting point as similar as possible to the desired configuration, the input SSA and COCO models are superimposed before each run, by performing the following operations:

1. rotate the SSA model about its longitudinal axis, causing its orientation to match that of the COCO pose estimate;
2. rigidly translate the SSA model, in such a way that the midway point of the segments connecting its hip joints overlaps the corresponding point in the 3D pose estimate.

The main goal of this approach is to alleviate the issues described in the previous section by inducing a more realistic placement of the 0^{th} and 11^{th} SSA joints. As can be seen from Figure 4.15, the articulated model's pelvis and hips are linked in a rigid triangle by segments $(0,5)$, $(0,6)$ and $(5,6)$, meaning the position of the first keypoint is heavily constrained by that of the other two. In practice, we found that when only taking the abdomen and legs into account (step 1 of the modified refinement procedure) an anatomically reasonable positioning of the pelvis could be obtained as a

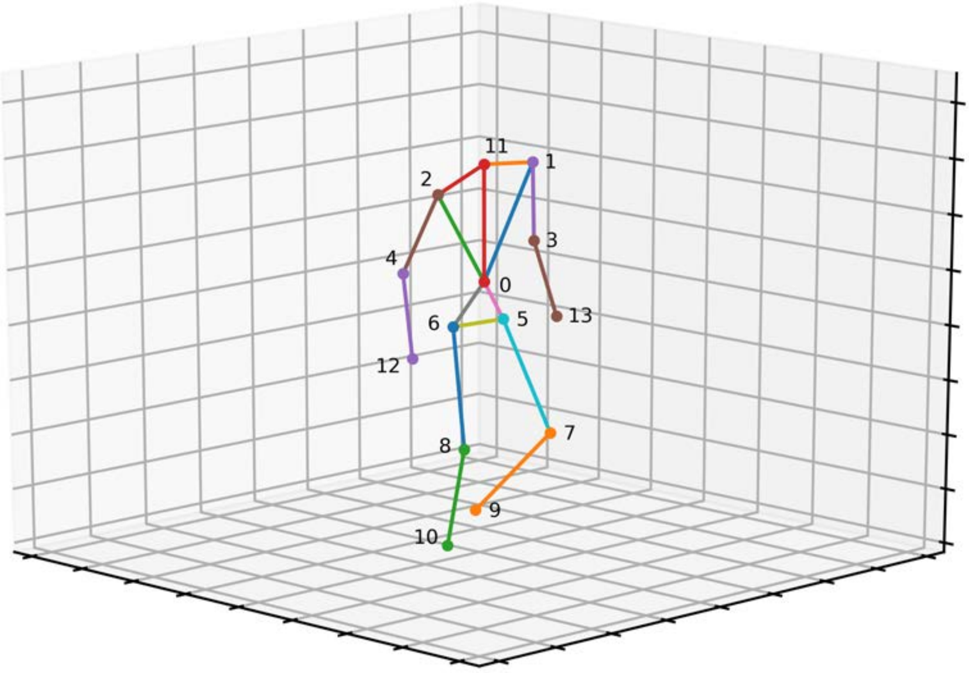


Figure 4.17: Example of an anatomically consistent pose, obtained by matching the articulated model shown in Figure 4.15 to the pose estimate displayed in Figure 4.13 in a two-step manner.

byproduct of matching the hip joints to their COCO counterpart. In the second stage, the determined lower-body keypoint locations are kept fixed, preventing constraints related to the torso from pushing the pelvis into incorrect positions. This in turn affects the optimization of the upper body, leading to a more realistic placement of the neck and shoulders.

Figure 4.17 displays the SSA configuration obtained by matching in a two-step manner the articulated model of Figure 4.15 to the 3D pose estimate shown in Figure 4.13. From a comparison with Figure 4.16, depicting the outcome of full-body matching on the same inputs, the two-stage approach can be seen to generate a significantly more natural pose. Furthermore, the modified procedure has the added benefit that in an underwater environment, in which the position of upper-body joints cannot be accurately triangulated, only the first step may be performed, substantially lowering the routine's runtime while still obtaining precise matches for the lower body.

4.4.3 Iterative Model Matching

In order to gather clinically meaningful information about the movement of a subject throughout a video sequence, the refinement procedure described in the previous section has to be applied repeatedly, matching the appropriate articulated model to a series of pose estimates (one per frame), each of which differs only slightly from its predecessor. When performing this type of iterative adjustments, it is common practice in the numerical optimization literature to employ the output of one iteration as the input of the next; for example, this approach is utilized in the MMC pipeline to facilitate the convergence of the model-to-VH registration algorithm [2]. Although we did experiment with this technique, we ultimately decided against its use in this work; instead, an articulated model in reference pose (such as the one shown shown in Figure 4.15) is independently matched at each iteration to the corresponding 3D pose estimate.

Much like the two-stage structure of the matching process, this choice was motivated by the need to obtain anatomically feasible placements of the SSA pelvis joint. In carrying out repeated OW matches, each taking as a starting point the configuration produced by the previous one, we found the 0^{th} keypoint would often take an incorrectly recessed position. Initially negligible, this effect would be amplified at each iteration to eventually become very significant (see Figure 4.18 for reference), at times also causing an unnatural displacement of the neck joint.

4.4.4 Considerations

Given that the main goal of this thesis was to develop a gait tracking methodology alternative to the one described in Chapter 2, it may seem counterintuitive for our matching approach to include elements (specifically the SSA model description) of the markerless motion capture pipeline. While some form of prediction refinement was necessary (due to the inaccuracy of the triangulated pose estimates), the decision to make use articulated models was motivated by convenience and availability considerations. Firstly, by employing in our procedure the same kinematic subject representation as MMC, we were able to draw more meaningful comparisons between the results produced by the two methods. Secondly, because the walking trials analyzed in this work had already been processed using markerless techniques in [2] and

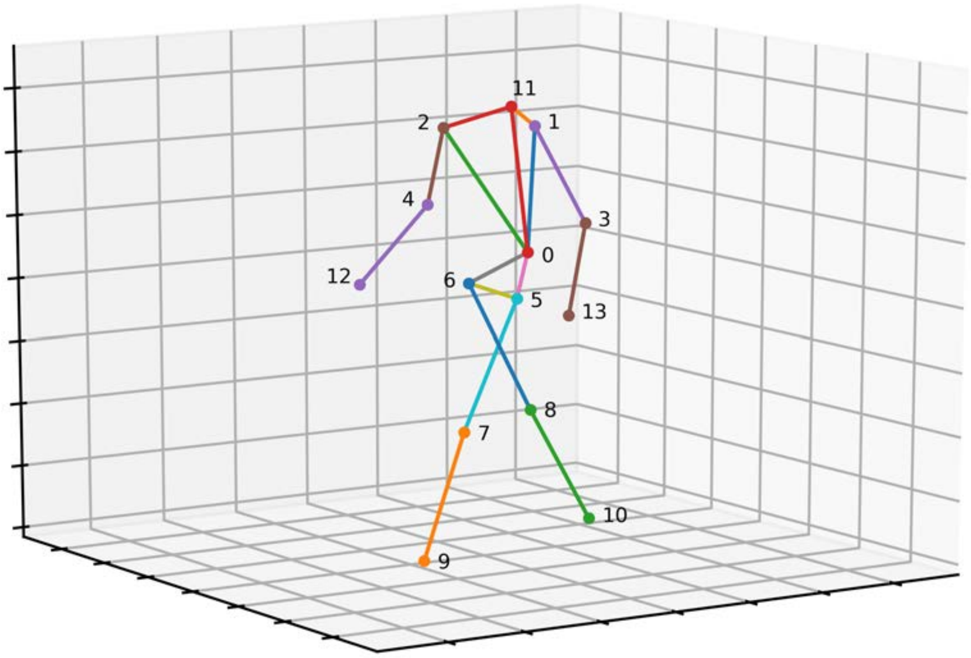


Figure 4.18: Unnatural articulated model configuration, obtained as the 38th element in a sequence of OW matches, each taking as input the output the previous one.

[3], suitable SSA models were readily accessible to us.

To conclude this section, we point out that although in this work we opted to employ the articulated model configuration shown in Figure 4.15, our proposed matching procedure is fairly flexible, and could be applied to different body representations with minimal modifications. In particular, it seems likely that an alternative model generation methodology, which does not rely on low-resolution visual hulls, could be developed using deep learning techniques, especially considering that CNN-based approaches collecting 3D shape information from multiple views of a human subject already exist in the literature [55, 72].

4.5 Joint Angles Calculation

The SSA models generated by the two-step matching procedure were employed for the calculation of anatomical joint angles. In the motion capture literature, this operation is usually carried out by embedding, in each considered body segment, an orthogonal frame of reference (X, Y, Z). A subject's motion throughout a video sequence may then be characterized by describing how the relative orientation of adjacent reference systems (RSs) changes over time.

Due to its reliance on morphological 3D shape information,[14] missing from the employed SSA representation, the RS definition technique utilized in [2] and [3] could not be replicated in this work. It was therefore necessary to devise an alternative approach, requiring only the kinematic information carried by a subject-specific articulated model, while still allowing meaningful comparisons to be drawn between our results and the ones produced by the MMC pipeline. A description of the technique developed for this purpose is given in the following paragraphs.

Orthonormal frames of reference were computed for six of the rigid segments considered in the SSA model, namely the abdomen (the triangle defined by keypoints 0, 5 and 6, with respect to the numbering convention adopted in Figure 4.15), the torso, the thighs and the shanks. Similarly to [2], upper-body limbs were not considered in our analysis; furthermore, due to the absence of feet in the SSA model, no joint angles could be computed for the ankles.

In the following, let:

- V_{ij} denote the vector going from the i^{th} SSA keypoint to the j^{th} (with respect to the numbering convention adopted in Figure 4.15), normalized to unit length;
- M_{ij} denote the midway point of the segment connecting the i^{th} and j^{th} joints;
- T_1 and T_2 denote the vectors going from the 0^{th} SSA keypoint (referred to as the pelvis) to, respectively, M_{56} and M_{12}.

The axes of the reference system (X_a, Y_a, Z_a), associated with the abdomen, were determined by performing the following assignments:

$$X_a = \frac{V_{56}}{\|V_{56}\|_2}, \qquad Z_a = \frac{X_a \times T_1}{\|X_a \times T_1\|_2}, \qquad Y_a = Z_a \times X_a. \tag{4.27}$$

Analogously, the torso's reference system (X_t, Y_t, Z_t) was defined by setting:

$$X_t = \frac{V_{12}}{\|V_{12}\|_2}, \qquad Y_t = \frac{Z_t \times T_2}{\|Z_t \times T_2\|_2}, \qquad Z_t = X_t \times Z_t. \tag{4.28}$$

[14] Provided in the MMC pipeline by the triangular mesh described in Section 2.2.

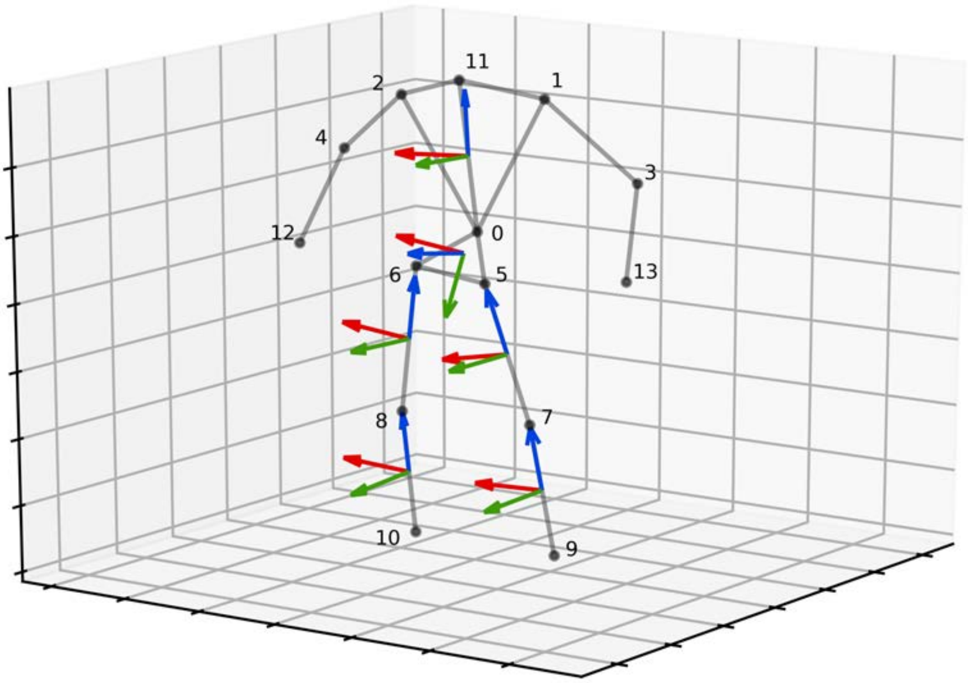


Figure 4.19: Representation of the reference systems associated with the considered SSA segments, in which the colors red, green and blue were used, respectively, to represent the X, Y and Z axes of each RF.

In all four remaining reference systems, embedded in the SSA model's femurs and tibias, the orientation of the Z axis was chosen to match that of corresponding limb, while Y and X were computed as:

$$Y = \frac{Z \times V_{56}}{\|Z \times V_{56}\|_2}, \qquad X = Y \times Z\,. \tag{4.29}$$

Figure 4.19 showcases the six frames of reference obtained from a static SSA model by the procedure described above. Note that, while for clarity the origin of each RS was placed in the center of the corresponding segment, this parameter is not considered in the calculation of joint angles, which only takes the orientation of adjacent reference systems into account.

Joint angles were computed for the hips and knees according to the approach proposed by Grood and Suntay in [73]. Given a pair of reference systems R_1 and R_2, embedded in the segments adjacent to the considered joint,[15] this procedure describes the orientation of the former with respect to the latter by means of three Cardan

[15] Abdomen and torso for the pelvis, abdomen and thigh for the hip, thigh and shank for the knee.

angles, referred to in the literature as:

- flexion/extension;
- abduction/adduction;
- internal/external rotation.

We refer the reader to [3, Section 2.3.1] for a detailed description of these angles' anatomical interpretation.

5
Results

5.1 Underwater Environment

The developed pipeline was applied to recordings of four male subjects, each of whom carried out six walking trials in an underwater environment. From each set of synchronized videos (recorded at 50 frames per second through the setup shown in Figure 4.2 (bottom)) we extracted a number of gait cycles (GCs) for both the left and right leg, which we then processed to obtain the flexion/extension angles associated with the hips and knees. Other reference systems, as well as rotation and adduction angles (shown not to be clinically meaningful when extracted through MMC techniques [74]) were not taken into consideration in this work.

Following an established practice in motion capture literature, the joint profiles obtained for each gait cycle were resampled to 100 data points, normalizing them to a common length and allowing a more meaningful comparison of their evolution: an FFT-based interpolation technique was employed for this purpose. A moving-average smoothing filter (having a window size of 5) was also applied to each trajectory, in order to reduce noise and remove possible outliers.

In agreement with previous publications [75], the trajectories computed for a given subject and joint across all GCs were employed to compute normative bands (mean ± standard deviation). The resulting profiles are showcased in Figures 5.1 to 5.16.

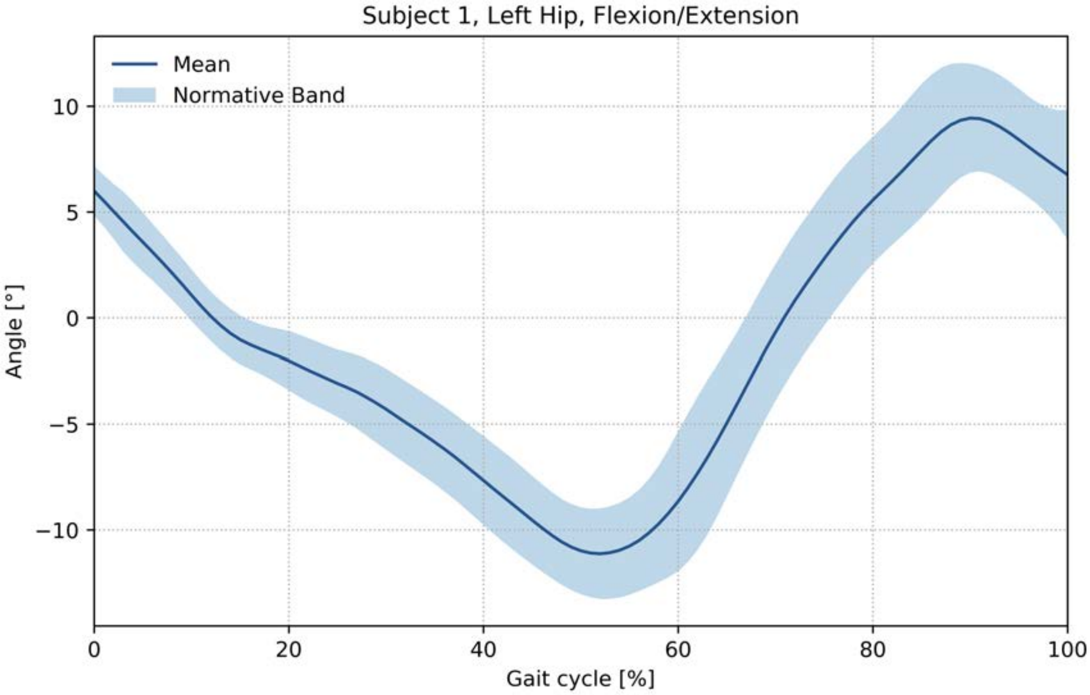


Figure 5.1: Subject 1: average joint profile of the left hip (± SD), computed over 13 gait cycles.

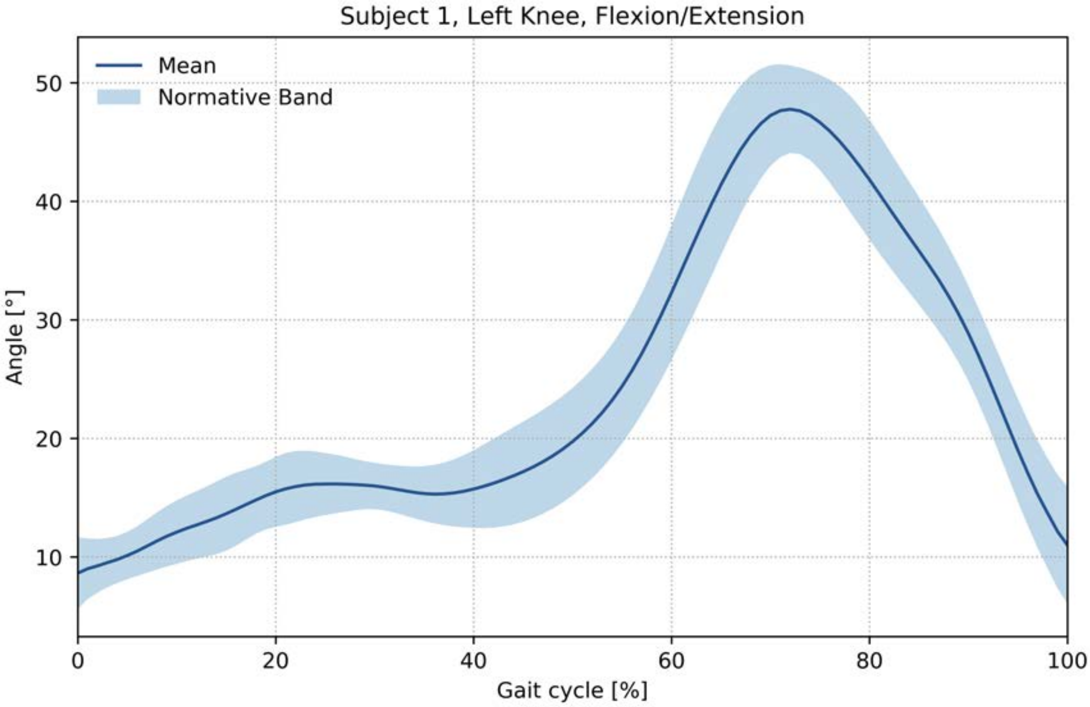


Figure 5.2: Subject 1: average joint profile of the left knee (± SD), computed over 13 gait cycles.

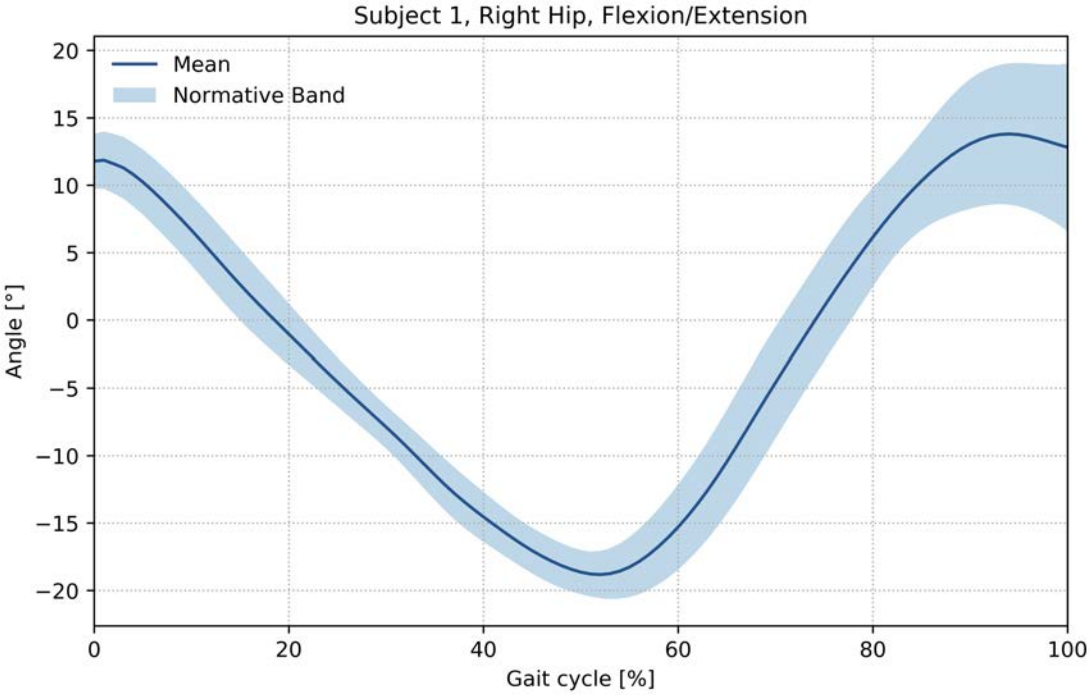


Figure 5.3: Subject 1: average joint profile of the right hip (± SD), computed over 12 gait cycles.

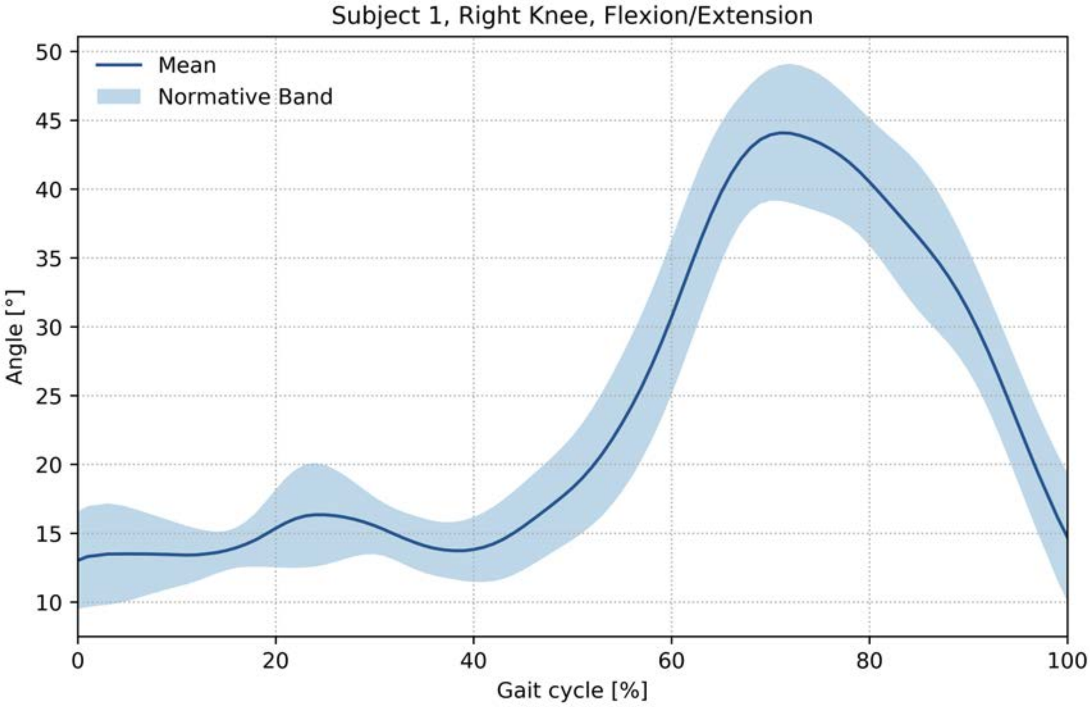


Figure 5.4: Subject 1: average joint profile of the right knee (± SD), computed over 12 gait cycles.

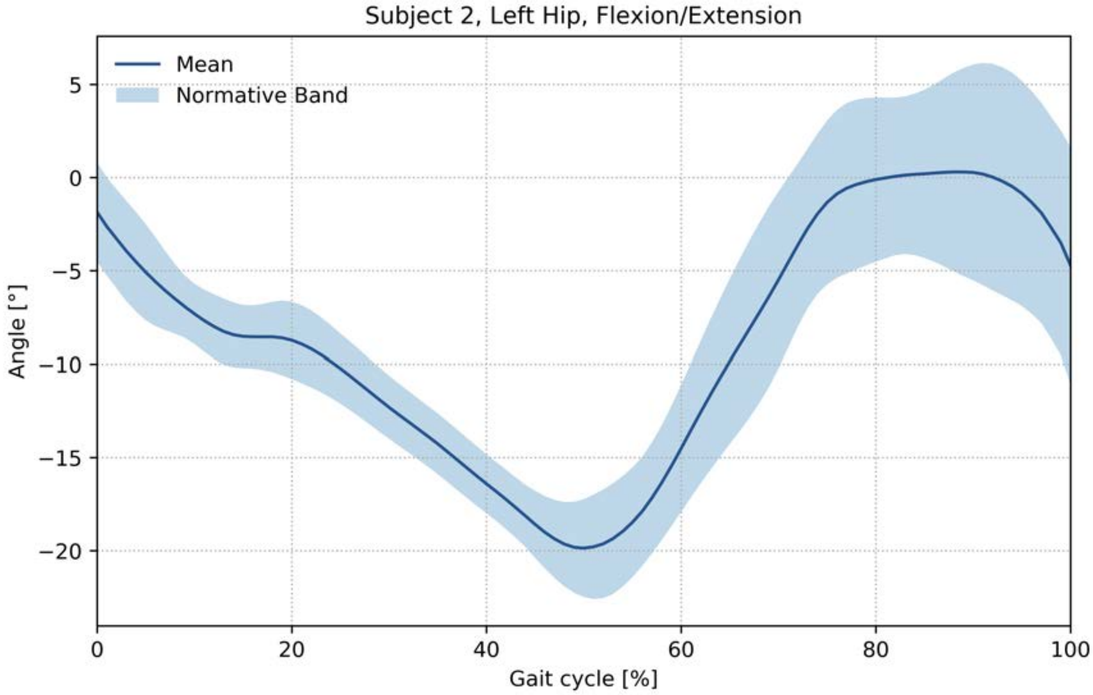


Figure 5.5: Subject 2: average joint profile of the left hip (± SD), computed over 12 gait cycles.

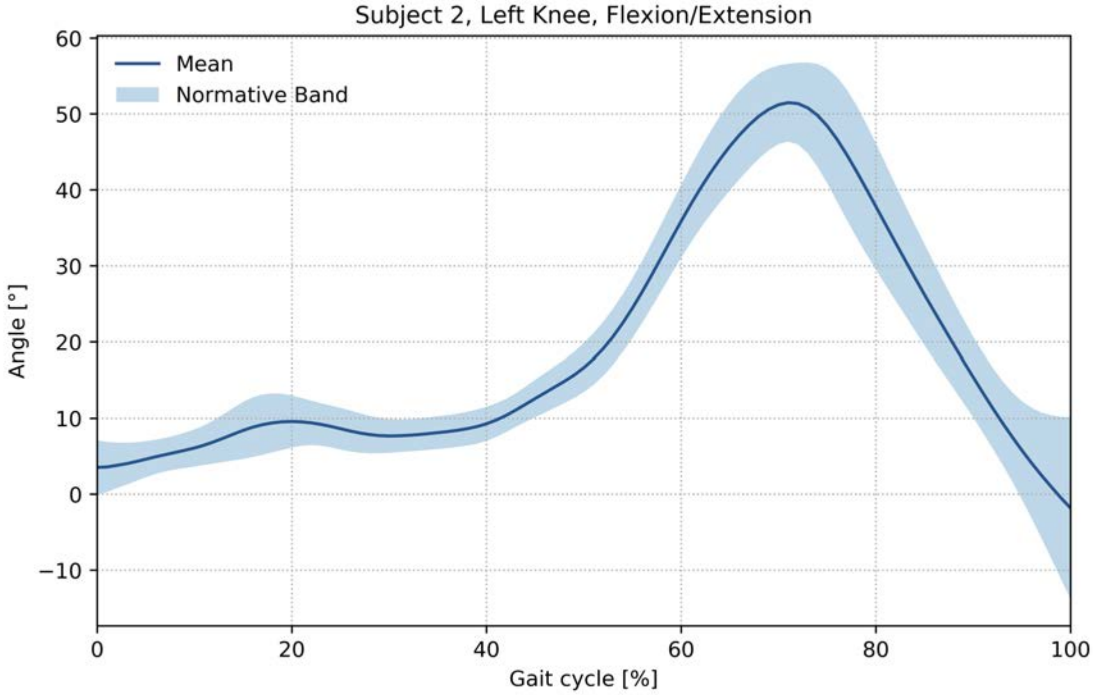


Figure 5.6: Subject 2: average joint profile of the left knee (± SD), computed over 12 gait cycles.

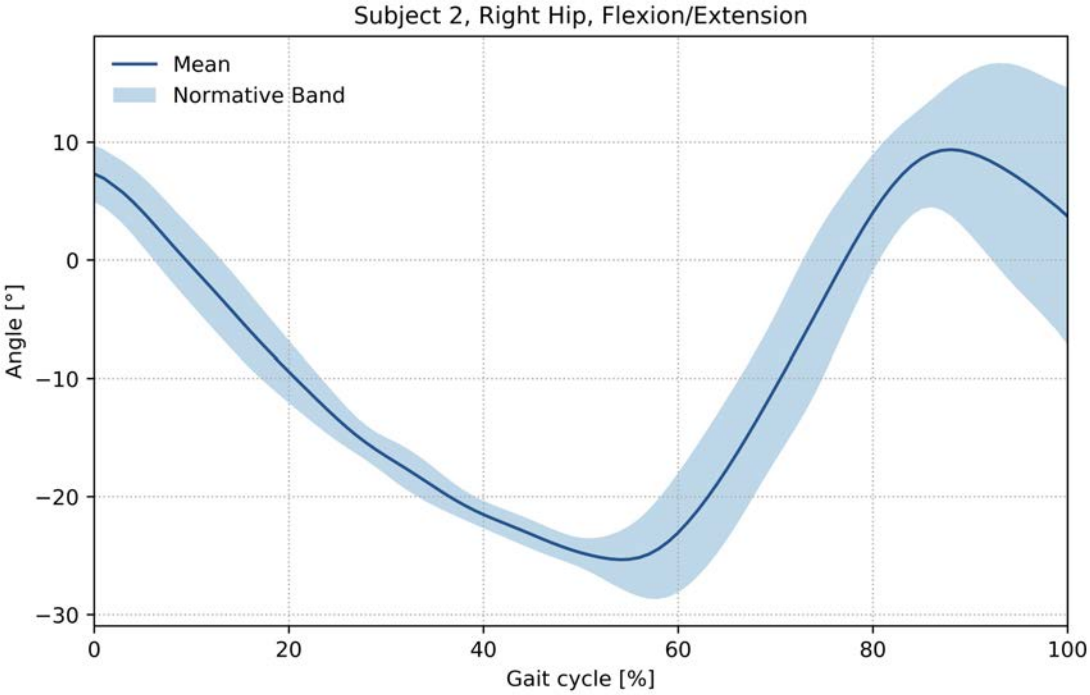


Figure 5.7: Subject 2: average joint profile of the right hip (± SD), computed over 12 gait cycles.

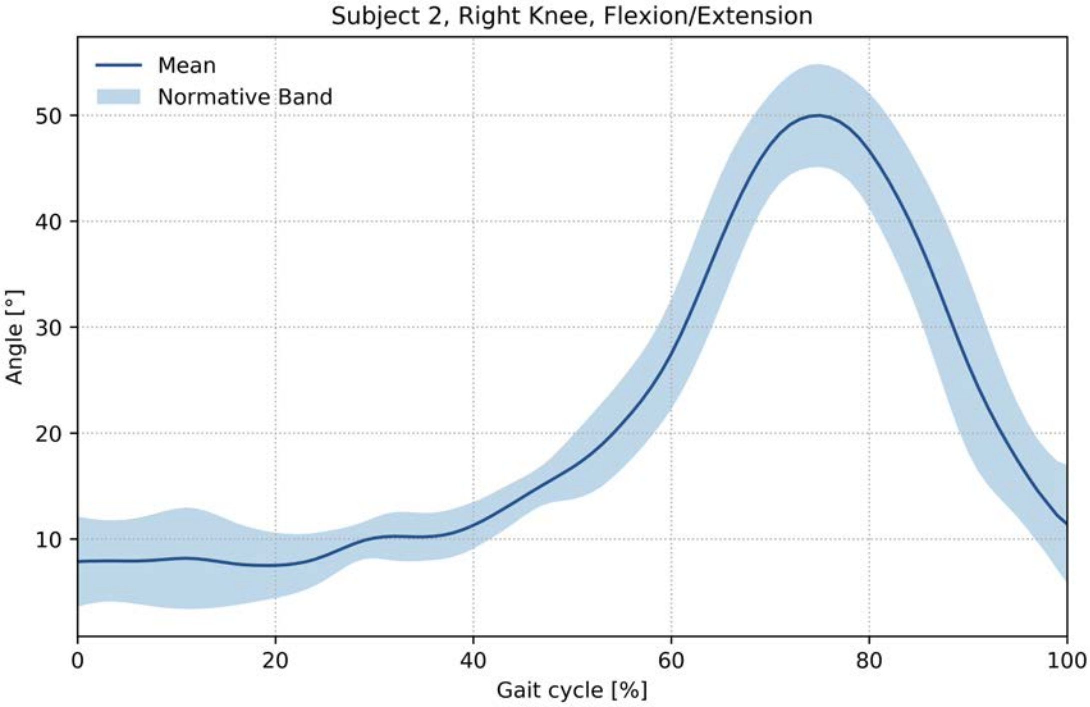


Figure 5.8: Subject 2: average joint profile of the right knee (± SD), computed over 12 gait cycles.

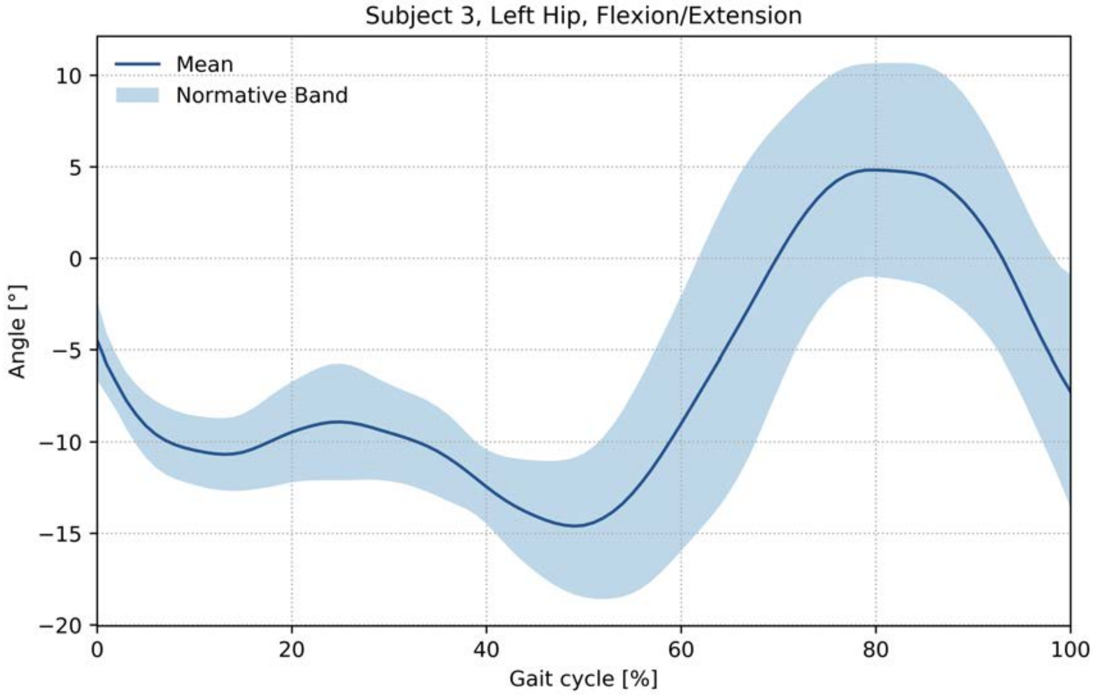


Figure 5.9: Subject 3: average joint profile of the left hip (± SD), computed over 15 gait cycles.

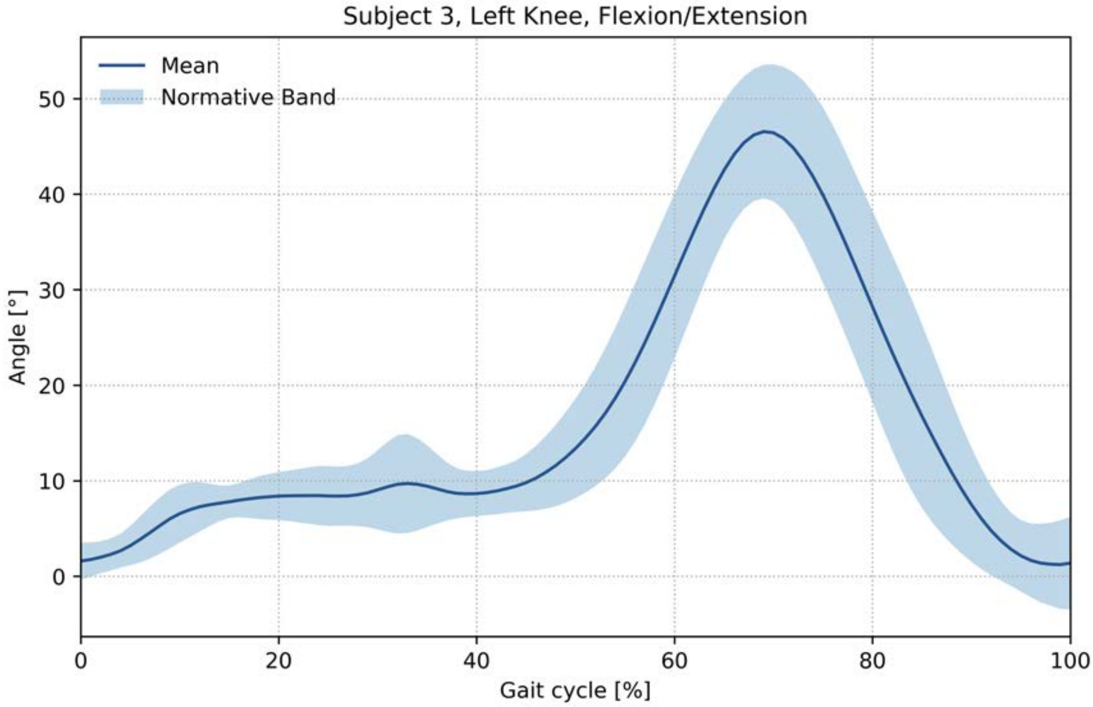


Figure 5.10: Subject 3: average joint profile of the left knee (± SD), computed over 15 gait cycles.

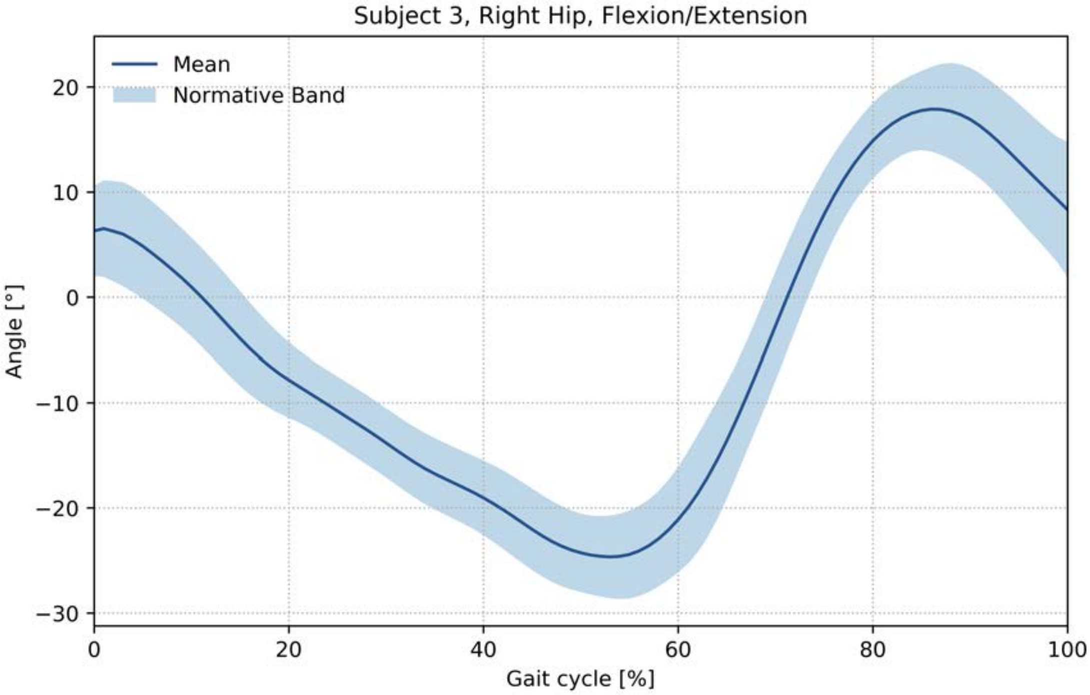


Figure 5.11: Subject 3: average joint profile of the right hip ($\pm$ SD), computed over 16 gait cycles.

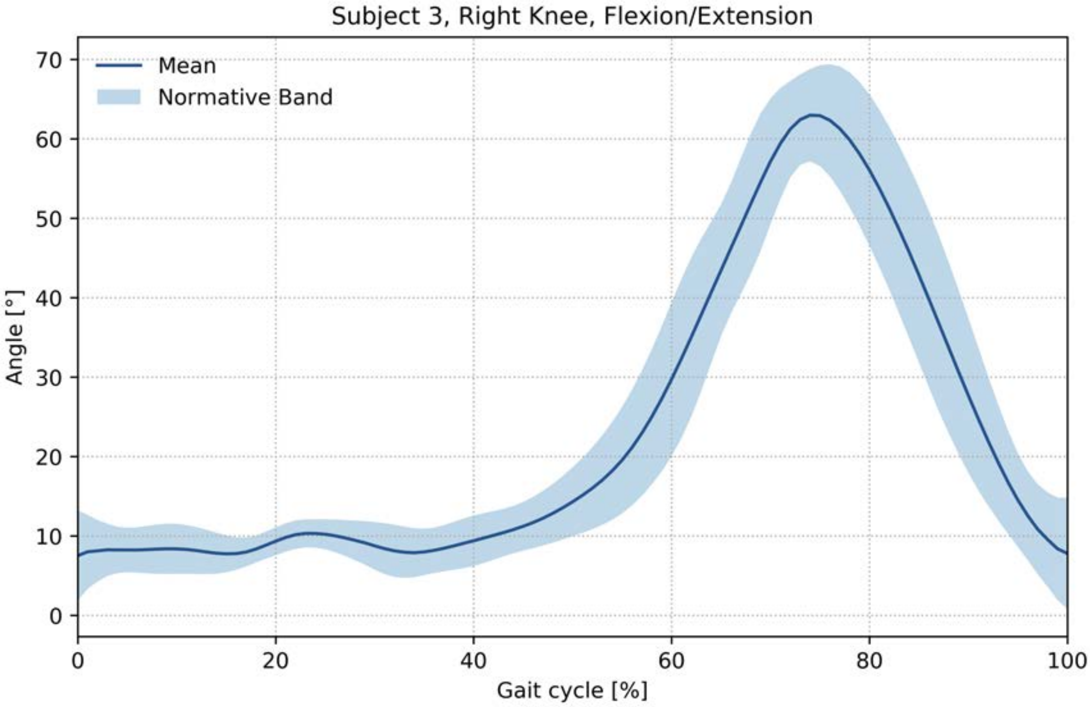


Figure 5.12: Subject 3: average joint profile of the right knee ($\pm$ SD), computed over 16 gait cycles.

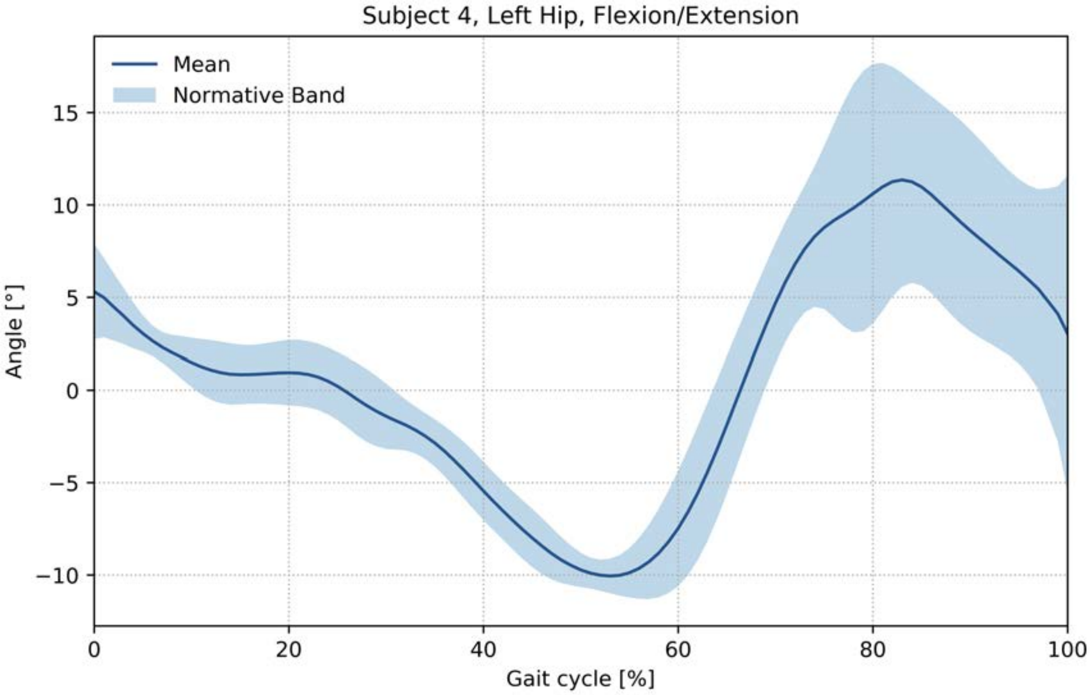


Figure 5.13: Subject 4: average joint profile of the left hip (± SD), computed over 10 gait cycles.

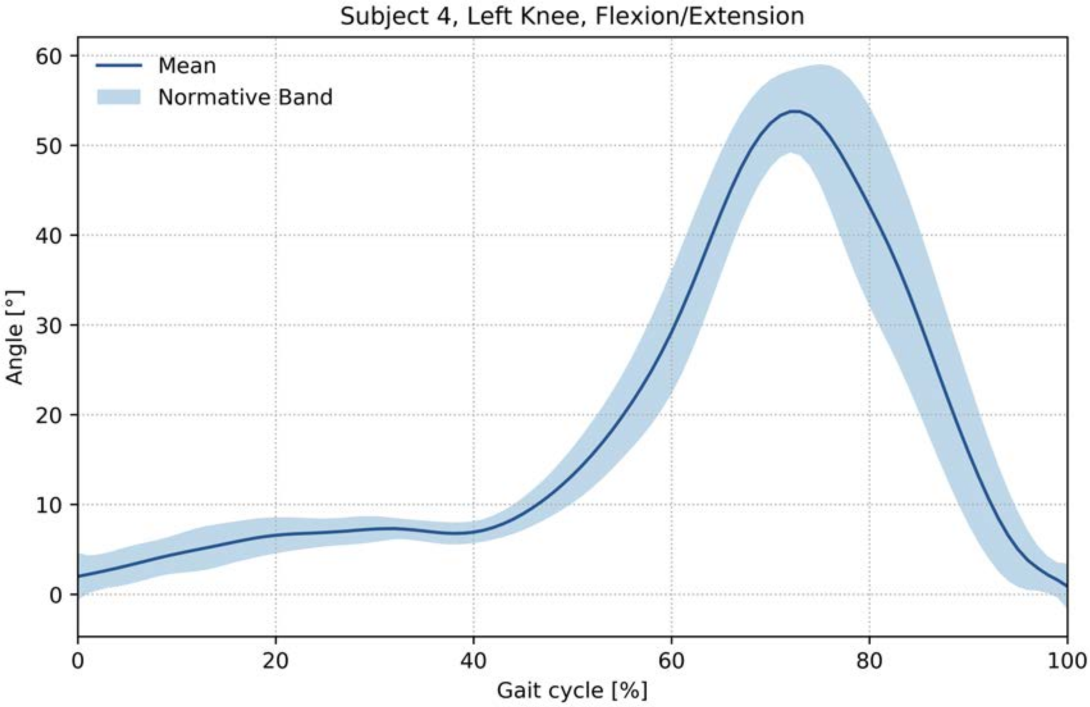


Figure 5.14: Subject 4: average joint profile of the left knee (± SD), computed over 10 gait cycles.

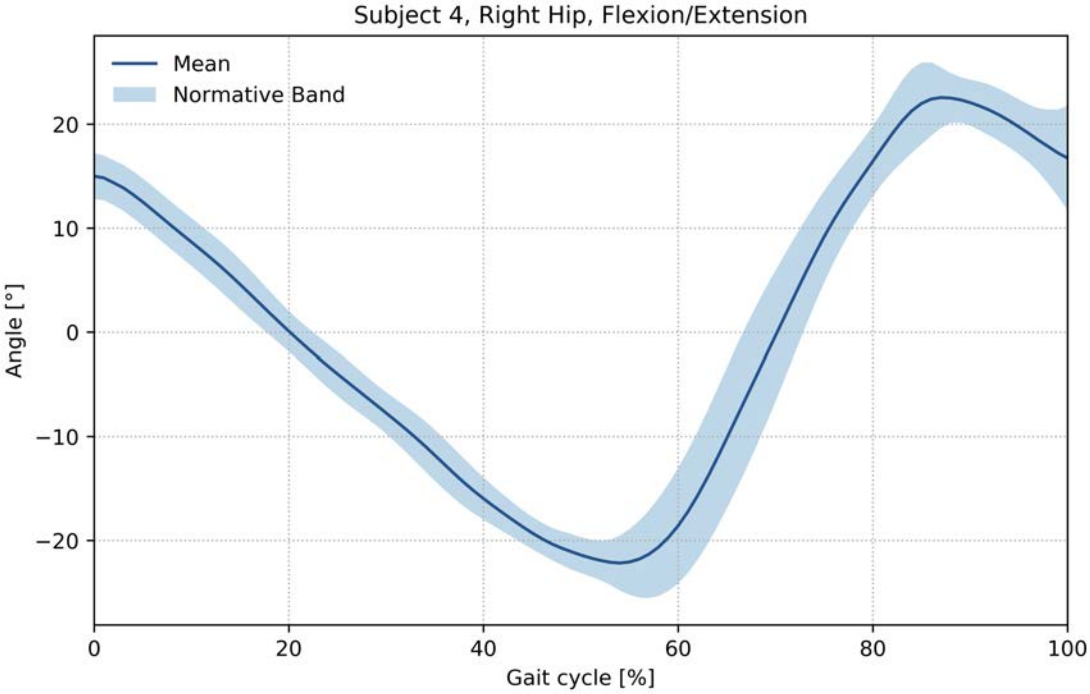


Figure 5.15: Subject 4: average joint profile of the right hip (± SD), computed over 10 gait cycles.

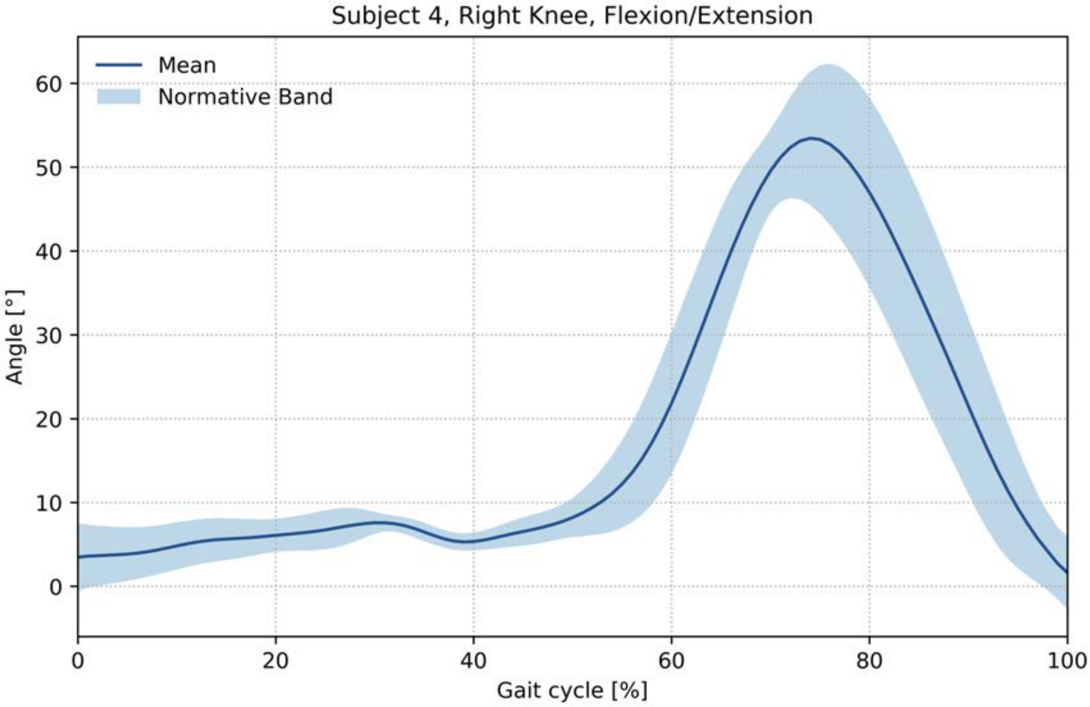


Figure 5.16: Subject 4: average joint profile of the right knee (± SD), computed over 10 gait cycles.

5.2 Out of Water Environment

Due to a lack of annotations concerning the beginning and end of GCs, only two sets of synchronized videos, containing views of a single subject[1], were available for elaboration in an OW environment. From each of these, we extracted two gait cycles (one for each leg), which we then processed to obtain the flexion/extension angles associated with the hips and knees.

Because OW trials were recorded at only 25 frames per second (as opposed to 50 in underwater conditions), as well as the fact that subjects tended to move faster when out of water, joint profiles computed in this environment were comprised of very few samples (25 to 28). In order to accommodate for this, slight modifications had to be made to the post-processing methodology described in Section 5.1.

Firstly, when upsampling to 100 data points, the previously employed FFT interpolation method was found to produce noticeable ringing artefacts, and was therefore replaced by an alternative approach based on linear splines.

Secondly, in order to prevent the interpolated samples from “amplifying” spurious predictions, a moving-average smoothing filter was applied both before and after the resampling operation.

Lastly, due to the small size of the test sample, we opted not to compute normative bands in an OW environment. Instead, the profiles obtained for each joint across all available GCs were shown, together with their mean, in Figures 5.17 to 5.20.

[1] Referred to as subject 2 in the previous section.

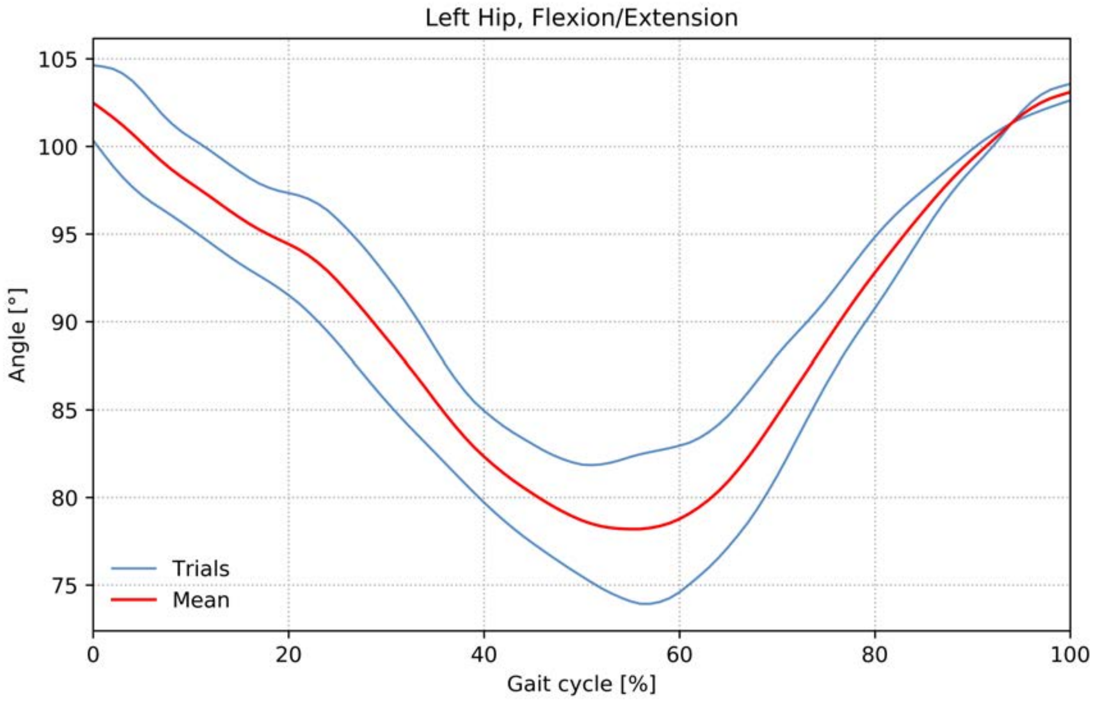


Figure 5.17: Subject 2: joint profiles obtained for the left hip.

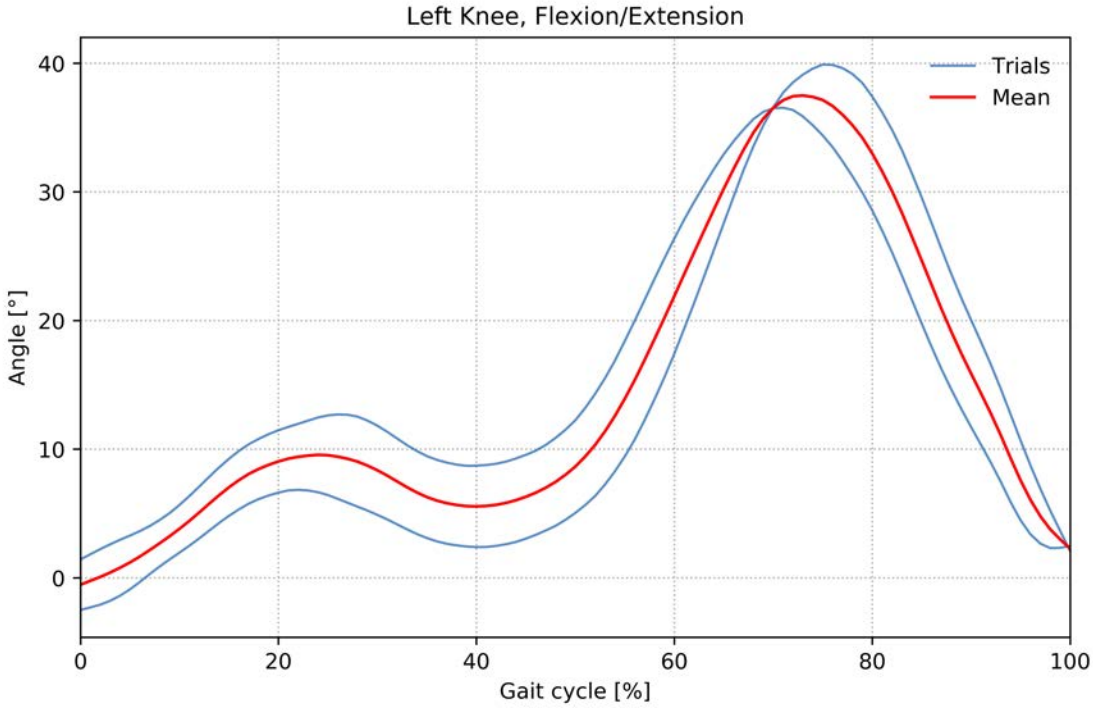


Figure 5.18: Subject 2: joint profiles obtained for the left knee.

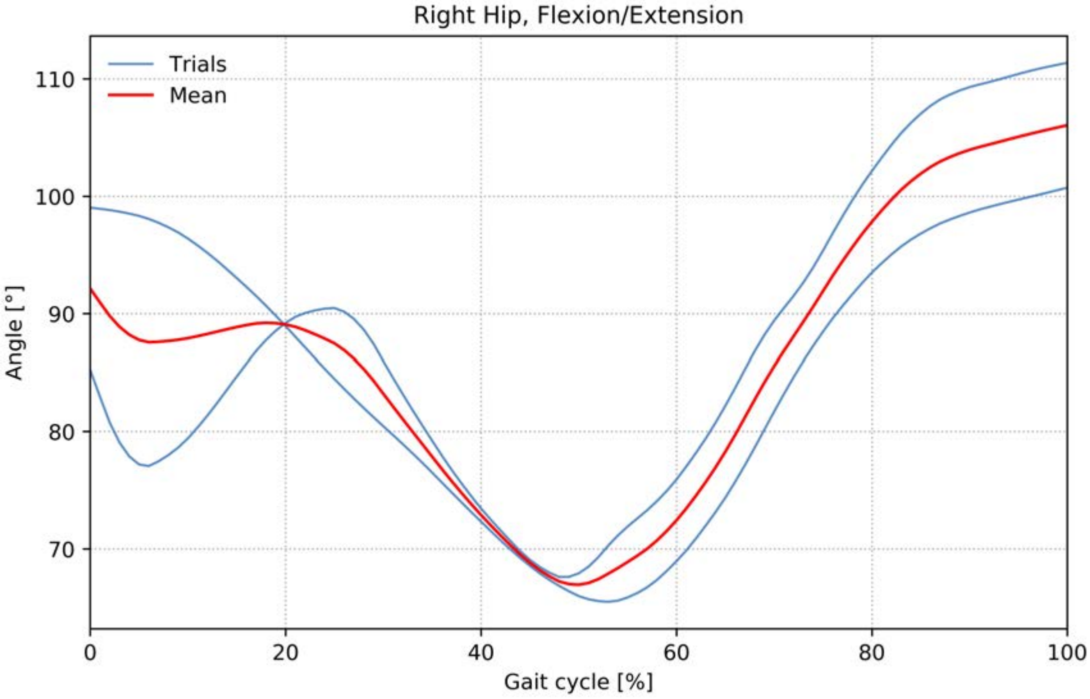


Figure 5.19: Subject 2: joint profiles obtained for the right hip.

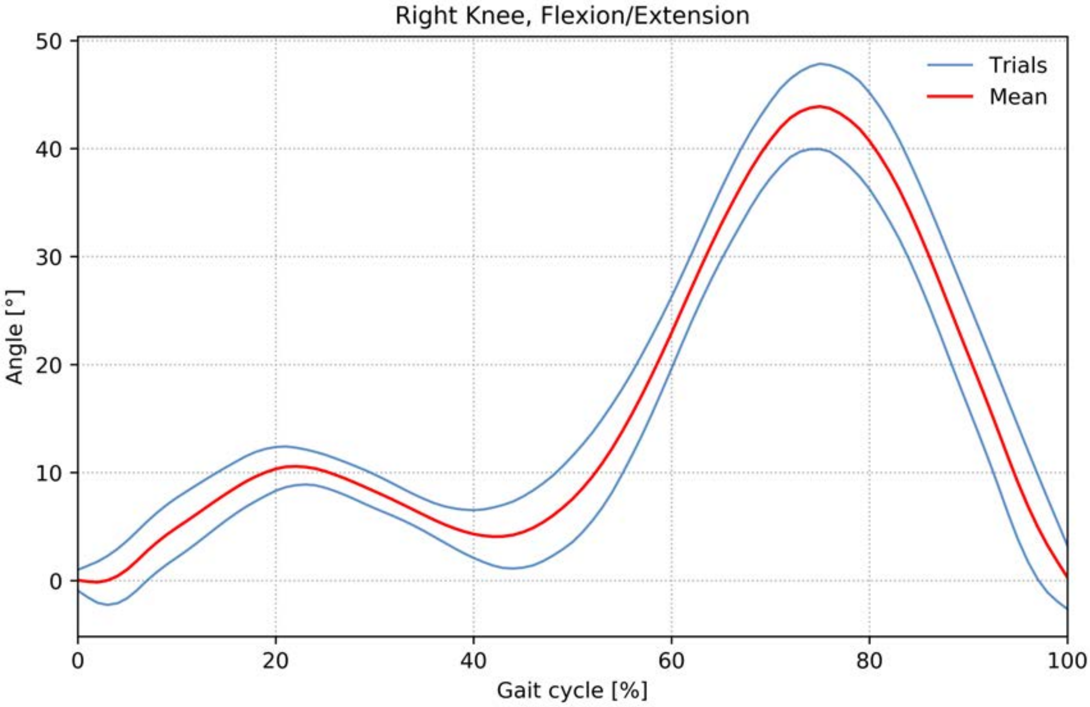


Figure 5.20: Subject 2: joint profiles obtained for the right knee.

6
Conclusions and Future Work

The main objective of the present thesis has been the development of a CNN-based markerless motion capture pipeline, aimed in particular at the analysis of a human subject's gait. Leveraging recent advancements in the field of pose estimation, our approach provides a research direction alternative to commonly employed MMC methodologies, which primarily rely on background subtraction [76] and visual hull reconstruction [74, 77]. Compared to other tracking techniques making use of neural networks [56, 57], a greater emphasis was placed on the retrieval of anatomically accurate movement information, via the implementation of a subject-specific prediction refinement routine.

The developed pipeline was applied to a dataset [75, 78] made available by the University of Padova's BioMovLab, containing recordings of walking subjects taken by six synchronized cameras, in both out-of-water and underwater conditions. Trials were selected for elaboration depending on the availability of a reference analysis, executed by means of a state-of-the-art markerless approach [2, 3]. This requirement was met by 24 UW walks (carried out by four subjects) and 2 OW ones (performed by a single individual), for each of which the joint profiles of the hips and knees were computed. Since the adduction and rotation angles produced by MMC techniques were shown not to be clinically meaningful [74], only angles lying on the sagittal plane were taken into consideration in this work.

The results produced by our system were found to be largely in agreement with

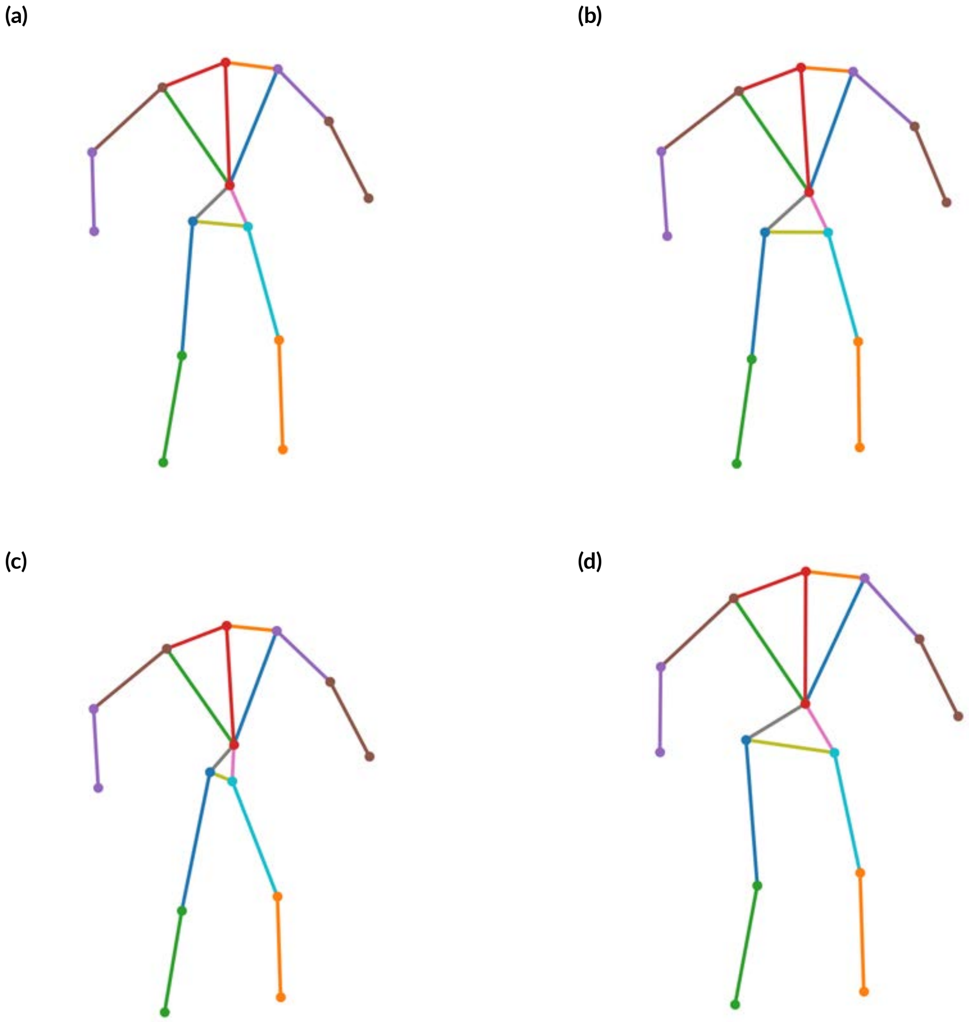


Figure 6.1: Articulated models associated with subjects 1 (a), 2 (b), 3 (c) and 4 (d).

previous publications [75], both in terms of the estimated joint profiles and ranges of motion, highlighting the viability of the proposed approach for the tracking of a human subject. Due to time constraints, we were however unable to provide a quantitative comparison between our approach and established gait analysis techniques, and thus could not assess whether the desired level of precision was achieved. In an underwater environment, we furthermore report the presence of significant inaccuracies in the reconstruction of the left hip's trajectory, particularly for subjects 2, 3 and 4. This behaviour is most likely owed to flaws in the employed subject-specific articulated models, which, as showcased in Figure 6.1, exhibit imperfections in the abdomen area (especially evident in (c)). These artefacts, caused by the garments worn by subjects during the models' definition, are believed to have negatively influenced the performance of our prediction refinement routine, resulting in the increased sample variance (and consequently wider normative bands) observed in Figures 5.5, 5.9 and 5.13.

Differently from other gait analysis studies [78], the flexion/extension angles computed for the pelvis were not reported in this work. This decision was motivated by a limitation discovered in the proposed model-matching algorithm, which, as discussed in Section 4.4.3, does not take the output of previous iterations into account when applied to a series of frames. This design choice, deemed necessary to prevent the method from reaching unnatural poses, had the undesired side effect of "locking" the orientation of the abdomen segment (the rigid triangle defined in Figure 4.15 by joints 0, 5 and 6), causing it to hardly vary throughout the elaborated sequences. As a result, the pelvis joint was found to rigidly shift along the subject's walking direction, rather than performing, as expected, an oscillatory movement concurrent to that of the hips.

The inability to accurately predict the position of the pelvis joint also prevented us from obtaining meaningful estimates for the configuration of the subject's torsos, whose trajectories were therefore not considered in this work. This difficulty was exacerbated in UW conditions by the presence of reflections in the water surface, which often occluded the subjects' shoulders and arms from view. Additionally, due to the absence in both the COCO and SSA models of keypoints denoting the feet, the angles associated with the subjects' ankles could not be evaluated.

In future work, we plan to provide a quantitative assessment of our system's performance, carrying out an exhaustive comparison between the results presented in Chapter 5 and the ones obtained, on the same dataset, by other gait analysis techniques, both markerless [74] and marker-based [75]. Among possible measures, the sample-wise Root Mean Square Distance (RMSD), employed in a several previous publications, is likely the most adequate. Furthermore, we intend to evaluate the proposed methodology on a wider test sample, applying it to a greater number of subjects and walking trials, particularly in out-of-water conditions.

In order to broaden the applicability of our approach, we will research ways in which to modify the model-matching algorithm to return anatomically feasible configurations of the abdomen segment, allowing the estimation of meaningful angles for the pelvis and (in an OW environment) the torso. Methods to recover the joint profiles associated with the subjects' ankles will also be investigated.

Lastly, we believe it worthwhile to explore whether subject-specific information could be embedded directly in the pose estimation step of our pipeline, by means of a fine tuning of the employed CNN. This operation would require a more extensive modification of our pipeline, possibly removing the need for a prediction-refinement step altogether.

References

[1] C. Tomasi and T. Kanade, “Detection and tracking of point features,” International Journal of Computer Vision, Tech. Rep., 1991.

[2] A. Mantoan, “Underwater gait analysis: a markerless approach,” 2011.

[3] F. Minelle, “Sviluppo di un metodo per confrontare l’analisi del cammino markerless e l’analisi del cammino marker-based in ambiente acquatico,” 2013.

[4] P. KaewTraKulPong and R. Bowden, “An improved adaptive background mixture model for real-time tracking with shadow detection,” in *Video-based surveillance systems.* Springer, 2002, pp. 135–144.

[5] T. Bouwmans, S. Javed, M. Sultana, and S. K. Jung, “Deep neural network concepts for background subtraction: A systematic review and comparative evaluation,” *Neural Networks*, 2019.

[6] A. Laurentini, “The visual hull concept for silhouette-based image understanding,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 16, no. 2, pp. 150–162, 1994.

[7] S. Corazza, L. Muendermann, A. Chaudhari, T. Demattio, C. Cobelli, and T. P. Andriacchi, “A markerless motion capture system to study musculoskeletal biomechanics: Visual hull and simulated annealing approach,” *Annals of biomedical engineering*, vol. 34, no. 6, pp. 1019–1029, 2006.

[8] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan, “Image-based visual hulls,” in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques.* ACM Press/Addison-Wesley Publishing Co., 2000, pp. 369–374.

[9] L. Mundermann, S. Corazza, A. M. Chaudhari, E. J. Alexander, and T. P. Andriacchi, “Most favorable camera configuration for a shape-from-silhouette

markerless motion capture system for biomechanical analysis," in *Videometrics VIII*, vol. 5665. International Society for Optics and Photonics, 2005, p. 56650T.

[10] S. Corazza, E. Gambaretto, L. Mündermann, and T. P. Andriacchi, "Automatic generation of a subject-specific model for accurate markerless motion capture and biomechanical applications," *IEEE Transactions on biomedical engineering*, vol. 57, no. 4, pp. 806–812, 2008.

[11] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "Scape: shape completion and animation of people," in *ACM transactions on graphics (TOG)*, vol. 24, no. 3. ACM, 2005, pp. 408–416.

[12] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[13] L. Mundermann, S. Corazza, and T. P. Andriacchi, "Accurately measuring human movement using articulated icp with soft-joint constraints and a repository of articulated models," in *2007 IEEE conference on computer vision and pattern recognition.* IEEE, 2007, pp. 1–6.

[14] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.

[15] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm." in *3dim*, vol. 1, 2001, pp. 145–152.

[16] J. J. Moré, "The levenberg-marquardt algorithm: implementation and theory," in *Numerical analysis.* Springer, 1978, pp. 105–116.

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

[18] S. Johnson and M. Everingham, "Learning effective human pose estimation from inaccurate annotation," in *CVPR 2011.* IEEE, 2011, pp. 1465–1472.

[19] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *CVPR*, 2019.

[20] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 466–481.

[21] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7103–7112.

[22] C.-J. Chou, J.-T. Chien, and H.-T. Chen, "Self adversarial training for human pose estimation," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2018, pp. 17–30.

[23] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7291–7299.

[24] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Transactions on computers*, no. 1, pp. 67–92, 1973.

[25] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International journal of computer vision*, vol. 61, no. 1, pp. 55–79, 2005.

[26] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1653–1660.

[27] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[28] M. Eichner, V. Ferrari, and S. Zurich, "Better appearance models for pictorial structures." in *Bmvc*, vol. 2, 2009, p. 5.

[29] Y. Yang and D. Ramanan, “Articulated pose estimation with flexible mixtures-of-parts,” in *CVPR 2011*. IEEE, 2011, pp. 1385–1392.

[30] Y. Tian, C. L. Zitnick, and S. G. Narasimhan, “Exploring the spatial hierarchy of mixture models for human pose estimation,” in *European Conference on Computer Vision*. Springer, 2012, pp. 256–269.

[31] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 315–323.

[32] B. Sapp and B. Taskar, “Modec: Multimodal decomposable models for human pose estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3674–3681.

[33] S. Johnson and M. Everingham, “Clustered pose and nonlinear appearance models for human pose estimation.” in *bmvc*, vol. 2, no. 4, 2010, p. 5.

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

[35] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, “Efficient object localization using convolutional networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 648–656.

[36] A. Newell, K. Yang, and J. Deng, “Stacked hourglass networks for human pose estimation,” in *European Conference on Computer Vision*. Springer, 2016, pp. 483–499.

[37] P. Burt and E. Adelson, “The laplacian pyramid as a compact image code,” *IEEE Transactions on communications*, vol. 31, no. 4, pp. 532–540, 1983.

[38] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[39] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention.* Springer, 2015, pp. 234–241.

[40] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[41] R. Collobert, K. Kavukcuoglu, and C. Farabet, “Torch7: A matlab-like environment for machine learning,” in *BigLearn, NIPS workshop*, no. CONF, 2011.

[42] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, “Joint training of a convolutional network and a graphical model for human pose estimation,” in *Advances in neural information processing systems*, 2014, pp. 1799–1807.

[43] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, “Deconvolutional networks.” in *Cvpr*, vol. 10, 2010, p. 7.

[44] V. Dumoulin and F. Visin, “A guide to convolution arithmetic for deep learning,” *arXiv preprint arXiv:1603.07285*, 2016.

[45] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele, “Deepcut: Joint subset partition and labeling for multi person pose estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4929–4937.

[46] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, “Deepercut: A deeper, stronger, and faster multi-person pose estimation model,” in *European Conference on Computer Vision.* Springer, 2016, pp. 34–50.

[47] S. Kreiss, L. Bertoni, and A. Alahi, “Pifpaf: Composite fields for human pose estimation,” *arXiv preprint arXiv:1903.06593*, 2019.

[48] M. Kocabas, S. Karagoz, and E. Akbas, “Multiposenet: Fast multi-person pose estimation using pose residual network,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 417–433.

[49] C.-H. Chen and D. Ramanan, "3d human pose estimation= 2d pose estimation+ matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7035–7043.

[50] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei, "Towards 3d human pose estimation in the wild: a weakly-supervised approach," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 398–407.

[51] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2640–2649.

[52] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt, "Vnect: Real-time 3d human pose estimation with a single rgb camera," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 44, 2017.

[53] L. Sigal, A. O. Balan, and M. J. Black, "Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *International journal of computer vision*, vol. 87, no. 1-2, p. 4, 2010.

[54] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1325–1339, 2013.

[55] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 945–953.

[56] M. Carraro, M. Munaro, J. Burke, and E. Menegatti, "Real-time marker-less multi-person 3d pose estimation in rgb-depth camera networks," in *International Conference on Intelligent Autonomous Systems.* Springer, 2018, pp. 534–545.

[57] J. Dong, W. Jiang, Q. Huang, H. Bao, and X. Zhou, "Fast and robust multi-person 3d pose estimation from multiple views," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7792–7801.

[58] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic, "3d pictorial structures for multiple human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1669–1676.

[59] J.-Y. Bouguet, "Camera calibration toolbox for matlab," 2001.

[60] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[61] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7310–7311.

[62] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

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

[64] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[65] R. I. Hartley and P. Sturm, "Triangulation," *Computer vision and image understanding*, vol. 68, no. 2, pp. 146–157, 1997.

[66] A. Fusiello, “Visione computazionale,” *Appunti delle lezioni. Pubblicato a cura dell’autore*, 2008.

[67] G. H. Golub and C. Reinsch, “Singular value decomposition and least squares solutions,” in *Linear Algebra.* Springer, 1971, pp. 134–151.

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

[69] T. E. Oliphant, *A guide to NumPy.* Trelgol Publishing USA, 2006, vol. 1.

[70] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, 2000.

[71] D. Kraft, “A software package for sequential quadratic programming,” *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt*, 1988.

[72] M. Kocabas, S. Karagoz, and E. Akbas, “Self-supervised learning of 3d human pose using multi-view geometry,” *arXiv preprint arXiv:1903.02330*, 2019.

[73] E. S. Grood and W. J. Suntay, “A joint coordinate system for the clinical description of three-dimensional motions: application to the knee,” *Journal of biomechanical engineering*, vol. 105, no. 2, pp. 136–144, 1983.

[74] E. Ceseracciu, Z. Sawacha, and C. Cobelli, “Comparison of markerless and marker-based motion capture technologies through simultaneous data collection during gait: proof of concept,” *PloS one*, vol. 9, no. 3, p. e87640, 2014.

[75] Z. Sawacha, F. Minelle, M. Cortesi, G. Gatta, C. Cobelli, and S. Fantozzi, “3d underwater gait analysis: Comparison among 3 different protocols,” *Gait & Posture*, no. 42, pp. S89–S90, 2015.

[76] A. Castelli, G. Paolini, A. Cereatti, and U. Della Croce, “A 2d markerless gait analysis methodology: validation on healthy subjects,” *Computational and mathematical methods in medicine*, vol. 2015, 2015.

[77] S. Corazza, L. Mündermann, and T. Andriacchi, “A framework for the functional identification of joint centers using markerless motion capture, validation for the hip joint,” *Journal of biomechanics*, vol. 40, no. 15, pp. 3510–3515, 2007.

[78] A. Mantoan, M. Cortesi, E. Ceseracciu, Z. Sawacha, S. Fantozzi, G. Gatta, and C. Cobelli, “Markerless gait analysis: An underwater pilot study,” *Gait & Posture*, no. 35, p. S4, 2012.