ECOLE
POLYTECHNIQUE
DE BRUXELLES

ULB

UNIVERSITÉ LIBRE DE BRUXELLES

Study of requirements for a low-cost system to estimate 3D position of human joints

using depth cameras and machine learning

Mémoire présenté en vue de l'obtention du diplôme
d'Ingénieur Civil biomédical à finalité spécialisée

Benjamin Hainaut

Directeur
Professeur Olivier Debeir

Co-Promoteur
Thomas Legrand

Service
LISA

Année académique
2022 - 2023

ULB UNIVERSITÉ LIBRE DE BRUXELLES

ULB UNIVERSITÉ LIBRE DE BRUXELLES

Exemplaire à apposer sur le mémoire ou travail de fin d'études,
au verso de la première page de couverture.

Réservé au secrétariat :	**Mémoire réussi***	**OUI**
		NON

CONSULTATION DU MEMOIRE/TRAVAIL DE FIN D'ETUDES

Je soussigné

NOM :
Hainaut
……

PRENOM :
Benjamin
……

TITRE du travail :
Study of requirements for a low-cost system to estimate 3D position of human joints using depth cameras and machine learning
……
……

AUTORISE*

~~**REFUSE***~~

la consultation du présent mémoire/travail de fin d'études par les utilisateurs des bibliothèques de l'Université libre de Bruxelles.

Si la consultation est autorisée, le soussigné concède par la présente à l'Université libre de Bruxelles, pour toute la durée légale de protection de l'œuvre, une licence gratuite et non exclusive de reproduction et de communication au public de son œuvre précisée ci-dessus, sur supports graphiques ou électroniques, afin d'en permettre la consultation par les utilisateurs des bibliothèques de l'ULB et d'autres institutions dans les limites du prêt inter-bibliothèques.

** Biffer la mention inutile*

Fait en deux exemplaires, **Bruxelles, le** 20/08/2023

Signature

** Biffer la mention inutile*

Formulaire version 04/2015

Résumé

Study of requirements for a low-cost system for 3D position estimation of human joints using depth cameras and machine learning.

Benjamin Hainaut, Master of science in Biomedical Engineering, Ecole polytechnique de Bruxelles, Université libre de Bruxelles, 2022-2023.

Ce mémoire a pour objectif d'étudier l'influence de différents paramètres à prendre en compte lors de la mise en place d'un système à bas-coût d'estimation de la position des articulations du corps humain en utilisant des caméras de profondeur et des réseaux neuronaux. Les paramètres étudiés sont : le nombre, le placement, la synchronisation et la résolution des caméras utilisées.

Un réseau neuronal dont la structure est inspirée de celle du réseau PointNet a été développé et une partie de l'ensemble de données "CMU Panoptic" a servi aux entraînements des différents modèles. Avant l'utilisation de l'ensemble de données, un grand nombre de prétraitements a été effectué afin de nettoyer et préparer les données.

Une première preuve de concept a été mise en place afin de prouver le potentiel de l'approche, suivie de quatre expériences. La première expérience étudie l'influence du nombre et du placement des caméras lors de l'entraînement des modèles. Les modèles entraînés avec plusieurs caméras ont en moyenne de meilleurs résultats. Ensuite est testée l'influence du nombre et de la position des caméras lors de l'utilisation des modèles. Il est observé l'importance de conserver une cohérence entre l'installation d'entraînement et celle d'utilisation finale.

L'influence de la synchronisation des caméras est ensuite étudiée en mettant en place une désynchronisation artificielle comprise entre +3 et -3 images de décalage. Une désynchronisation légère est bénéfique lors de l'entraînement des modèles et n'a pas d'influence visible lors de l'utilisation des modèles. La dernière expérience étudie l'influence de la définition des caméras au travers la taille des nuages de points et détermine qu'une haute définition de l'entraînement est importante et que la résolution nécéssaire lors de l'utilisation est limitée.

Des travaux additionnels sont nécessaires pour totalement définir les paramètres nécessaires à la mise en place d'un système à bas-coût d'estimation de la position des articulations du corps humain en utilisant des caméras de profondeur et des réseaux neuronaux, de plus certaines limitations ont limité les résultats de ce mémoire et peuvent être résolus.

Mot-clés : Point cloud, Human joints, Neural network, PointNet, Pose Estimation, Depth Camera

Abstract

This thesis aims to investigate the influence of various parameters to be considered when implementing a low-cost system for estimating the positions of human body joints using depth cameras and neural networks. The studied parameters are the number, placement, synchronization, and resolution of the used cameras.

A neural network structure inspired by the PointNet architecture was developed, and a subset of the "CMU Panoptic" dataset was used to train various models. Prior to dataset utilization, extensive preprocessing was conducted to clean and prepare the data.

A preliminary proof of concept was established to demonstrate the potential of the approach, followed by four distinct experiments. The first experiment delves into the impact of camera number and placement during model training. Models trained with multiple cameras exhibited superior average results. Subsequently, the influence of camera count and position was tested in the context of model usage. The importance of maintaining coherence between the training setup and the usage setup was observed.

The impact of camera synchronization was then examined by introducing artificial desynchronization ranging from +3 to -3 frames offset. Slight desynchronization proved beneficial during model training and showed no significant influence during model usage. The final experiment studies the effect of camera resolution through point cloud size, determining that high-resolution training is crucial while usage resolution can be constrained.

Further endeavors are required to comprehensively define the parameters necessary for establishing a low-cost system for estimating human body joint positions using depth cameras and neural networks. Additionally, certain limitations have constrained the outcomes of this thesis, and potential resolutions for these limitations are identified.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter summary: In this chapter, we introduce the problem, define the goal of the thesis, and explain its structure.

Motion analysis is and will always be an essential aspect of kinesiotherapy, physiotherapy and osteopathy. Modern systems used to perform such an analysis are limited to large rooms with expansive set-ups and tedious procedures to prepare the subject and are therefore not used for routine tests. The state-of-the-art system used in motion capture is the Vicon system, which uses many cameras, requires markers placed on the subject, and costs at least a few thousand dollars for the smallest versions. The system also requires a dedicated room and is difficult to use outside the lab. There exist other solutions, but they are often either not cheaper or not as accurate.

Obtaining a stable, portable and cheap system that gives precise results in motion analysis could add needed quantitative information to practitioners, but is difficult. This thesis will study the requirements of a mobile, affordable, and easy-to-set-up system using depth cameras to estimate the articulation position in the human body accurately. The specificity of our approach is to work with a neural network approach using the fused point clouds of the human body coming from different synchronized Time of Flight cameras. This allows us to work directly with the depth data of all cameras at once. More precisely, we will focus on the following parameters of the capture system :

- Influence of the number of cameras
- Influence of the positioning of the cameras
- Influence of the good synchronization of the cameras
- Influence of the resolution of the cameras

In chapter 2, we will first rapidly pass over a history of motion analysis. We will then continue with a review of different methods used in motion analysis, starting with manual processes and methods using passive or active markers. We will then pass on to the markerless techniques using single 2D or multiple 2D or 3D cameras. We will then describe what is a Neural Network and its hyper-parameters.

In chapter 3, we will describe the method of the work of this thesis. We will start with the description of the network used, continue with the dataset choice and presentation, as well as the preprocessing of the dataset and the data management. Finally, we will rapidly describe the hardware used, our skeleton model and our implementation method.

In chapter 4, we will describe our experiments, starting with a proof of concept of our approach, followed by our different experiments and their results.

In chapter 5, we analyze and discuss the obtained results of our method and compare the global result with the state of the art. We then take a look at the observed limitations of our experiments;

Finally, in chapter 6, we give a conclusion on the results of our work and explore different possibilities available to go further in the future.

Chapter 2

State Of The Art

Chapter summary: This chapter is separated into three major parts: the first part passes quickly on the history of motion analysis, the second part reviews different systems used in motion analysis, with or without markers, and the last part presents the workings of a neural network.

2.1 Before the advent of computers

Since ancient times, medicine has been an essential part of human societies. The first traces of conscious medicine dates back to nearly 50 thousand years in a Neanderthal population in Spain [53]. We have to wait much longer to get the start of the movement analysis, more precisely, the walk analysis, known as gait analysis. We will rapidly get through the different breakthroughs in this domain, using the work of Richard Baker [3]

The first written occurrence of study of the human movement is much more recent with Aristotle and his fundamental analysis of the human walk in his work "Movement of Animal" in 350 BC [41]. It was necessary to wait for sciences and mathematics to develop further in Europe to get new advancements in movement analysis. The gait analysis was further developed over two centuries, but still limited because depending on instantaneous observations.

Modern technologies were then brought to this field at the end of the 19^{th} century, using the first cameras to capture the movement, annotate images and be able to study the movement in more detail[8]. An idea of the result of the capture can be seen in Figure 2.1

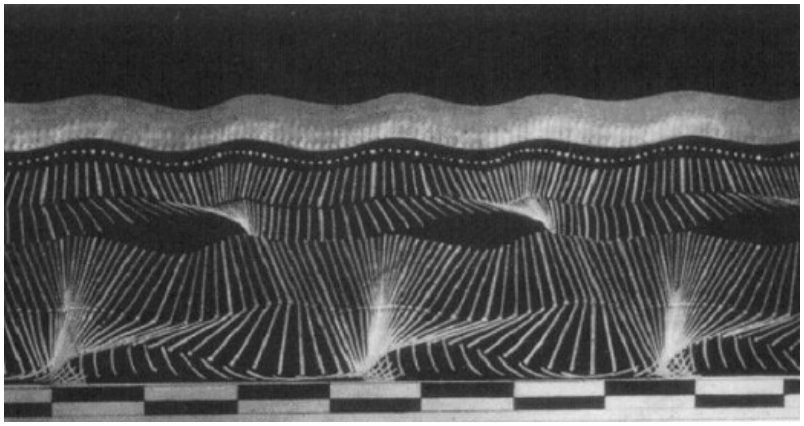


Figure 2.1: Jules Etienne Marey, Joinville Soldier Walking, 1883, geometric chronophotograph (Paris College de France)

This approach still had a big default; all the setup (a suit with lights) hindered the subject's natural movement. We will see further that this issue is still a problem in recent years and one of the drivers in the domain of movement analysis. Due to the long time required to prepare the subjects and the tedious task of processing the data obtained by hand, gait analysis and, globally, movement analysis were limited to a very small number of patients before the advent of computers.

2.2 Methods with body equipment

2.2.1 Manual annotation

With the advent of digital cameras, film-based cameras were slowly replaced; indeed, digital cameras' quality and frame rate rose, while their price diminished.

Images still needed to be annotated manually, and with the augmentation of the frame rate, and thus post-processing time, it was required to find a solution to ease this task. Different tools were developed for this purpose, giving estimations of the positions of points relatively precisely [54]

While tedious, this method is still sometimes used because of its advantages. Indeed, we do not require to equip the subject with markers that could hinder its movements, and the studied movement or task can be done outside of the controlled environment of the laboratory.

This method gives excellent results when performed by trained and experienced experts but keeps the drawbacks of being time-consuming and sensitive to human subjective errors. To solve this problem of time-consuming manual labeling, systems were developed to capture the subjects' movements automatically. They use different approaches that we will further discuss in this section.

2.2.2 Marker-based systems with inactive markers

A common solution is marker-based systems with inactive markers. Such systems have many implementations, but the main idea is always the same. We do not have a preference between the different actors in the market (Vicon, Qualisys, Nokov, ...), but we will focus on the Vicon system in this section.

It is based on three main components: an array of high frame rate cameras placed around the capture area, markers placed on the observed subject and a program to track the markers. We can see an example of such a setup in Figure 2.2

The cameras are high resolution and high frame rate infrared cameras, going from 8 megapixels and 500 frames per second to 26.2 megapixels and 150 fps. Around the objective is an infrared emitter. The number of cameras can range from 4 in the smaller setups to 60 or more in the bigger ones.

The passive markers are simple small reflective balls. They are placed on the subject at the different points of interest, either directly glued to the skin or via a bodysuit. In general, there is one marker on each point of interest, as shown in Figure 2.3 (a). Sometimes, single markers can be replaced by marker clusters. For one point of interest, we will put a set of markers at different positions and sometimes on a rigid support, as shown in Figure 2.3 (b). Those clusters have two

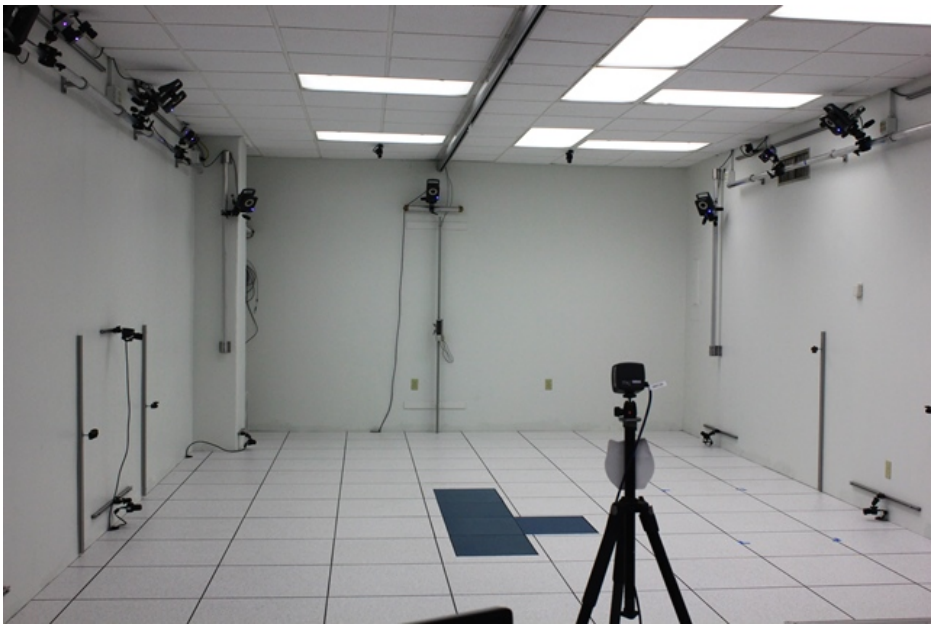
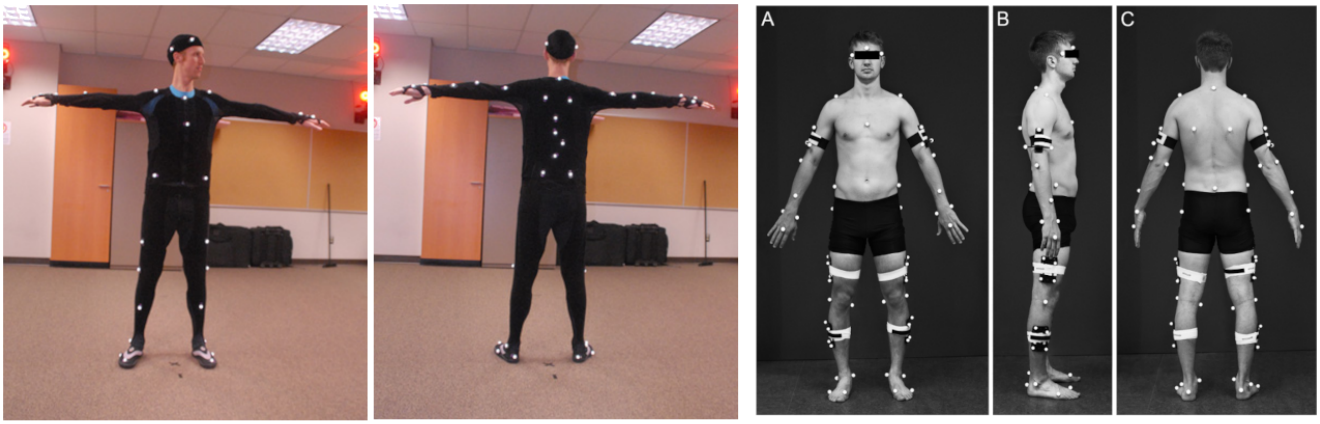


Figure 2.2: Mizzou Motion Analysis Center's Vicon setup [1]

advantages: they reduce the chance of losing the marker in all camera views simultaneously and reduce the perturbations due to skin movement.



(a) (b)

Figure 2.3: (a) Simple markers on a body suit, from Chris Joslin - School of Information Technology, Carleton University, Ottawa, Ontario, Canada. (b) Marker clusters, from "Explaining the unique nature of individual gait patterns with deep learning" by Horst and Al.[12]

Those kinds of systems are exact in tracking the markers, statically or in movement, having a global accuracy error inferior to two millimeters, as shown by Windolf and Al. in their study of the accuracy and precision of video motion capture systems [55]. Still, the quality of the result is always dependent on the calibration of the setup.

While the tracking of the markers is top-notch, those are not the final goal of the capture. The next step is thus to use the position of those markers to infer the position of the articulations and use this in our application (model animation, motion analysis, ...).

[1]Image taken from the Mizzou Motion Analysis Center's website on the 06/06/2023 (https://mizzoumotioncenter.com/facilities.html

Marker-based systems with passive makers are to this day one of the best solutions in terms of quality of results, but they come with a series of downsides that are important to improve upon:

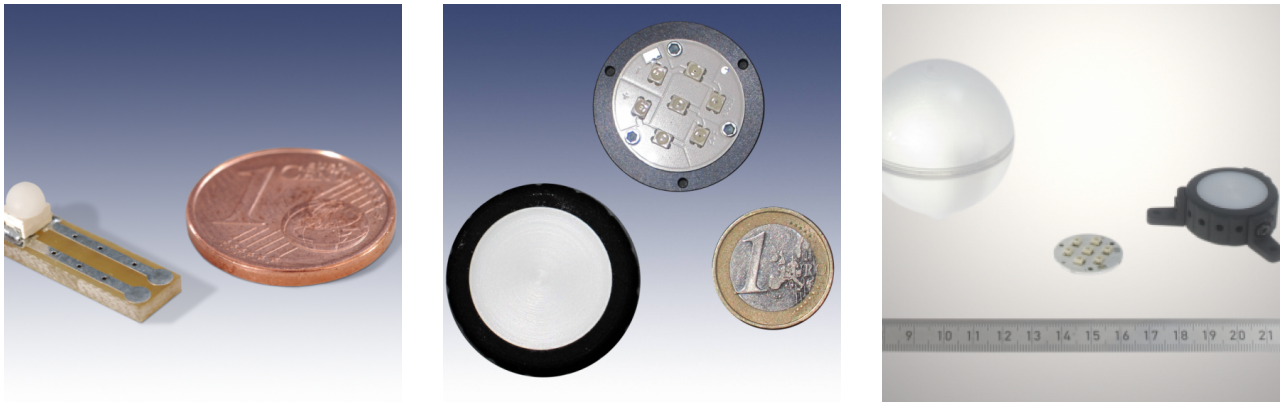
- **Occupied Space**: These systems require a dedicated room. The smallest setups require at least 12 square meters. Additionally, it is often needed to keep additional space for the structures supporting the cameras and for the rest of the hardware.
- **Cost**: These systems are very costly, starting at more than ten thousand euros for the smallest setups and going to astronomical amounts for the biggest ones. We need to take into account the price of the cameras, the markers and the different accessories, the hardware, the formation and the required software.
- **Time-consumption**: The system's initial setup is long, but if it is done correctly, it's only a one-time thing. However, whenever a subject comes, we must equip him with the markers. The placement is critical to get correct results, so it must be done meticulously. Additionally, the acquired data sometimes has to be manually corrected to assign every marker the right name and to be sure that there is no exchange of those markers during the whole capture.
- **Precision**: The system does not track the human articulations but the markers placed on the subject. Those markers can be subject to movements that do not correspond to the actual articulation's movements. Indeed, the markers can move with skin displacement and sometimes have translation movement additionally to a rotation movement, when the actual articulation only rotates as shown by Cappozzo et al. in "Position and orientation in space of bones during movement: experimental artifacts" [6].
- **Movement modification**: In a clinical setup, the objective is to study the movement of the subject, but due to the markers placed on his body, the subject's movement could be impeded (whether physically or on a more psychological level) and thus not reflect the real movement. It is then difficult to distinguish between a pathological factor and the system's influence on the patient.
- **Environment dependent**: These systems need a well-controlled environment to work correctly. It is mandatory not to have any reflective surfaces in the working space of the cameras as it would be interpreted as a marker, and the light has to be well controlled as well. It is thus nearly impossible to use this kind of system in a real environment.
- **Operator**: These systems require a trained person to place the markers and operate the system

Other systems exist and each has advantages and disadvantages and can be used as an alternative depending on the problem studied. Some of these systems are reviewed in the rest of the section.

2.2.3 Marker-based systems with active markers

Marker-based systems with active markers are very similar to the ones with passive markers. Obviously, the difference is that, in this one, the markers are active. They emit light at a specific frequency. This gives the system the advantage of having no risk of marker swapping. Indeed, with passive markers, if they overlap, the tracking software may exchange them. Here, every marker is uniquely defined by a specific frequency or blinking pattern. The tracking is then automated and the technician does not need to relabel markers throughout the capture. Another advantage is that these systems are more robust to environmental conditions and are thus easier to use outside of the laboratory.

There are, however, additional negative aspects: the weight and the cost. Active markers require a battery and some electronics and are thus heavier, which can interfere more with the subject's natural movement and cost a lot more. Depending on the money spent, these factors can be alleviated with smaller but more expansive markers. Different types of active markers are shown in Figure 2.4.



(a) Small active marker (b) Medium active marker (c) Big active marker

Figure 2.4: Pictures of different active markers sold by Advanced Realtime Tracking

This system does not solve the problems of occupied space, cost, imprecision and movement modification and only partially solves the problem of time consumption.

2.2.4 Intracortical markers

When the exact movement of the skeleton is needed, all skin and muscle movement needs to be eliminated. To do so, instead of placing the markers on the skin, they are directly linked to the bones as shown in Figure 2.5[31]. This allows the skeleton to be accurately tracked, but the setup could once again disturb the movement. However, Maiwald and his team showed in 2016 that those perturbations are limited and a valid gait analysis can still be performed. [28]

Figure 2.5: Skin marker directly on the skin (left), on a rigid plate (center) and attached to the bones (right) [31]

While this method allows highly accurate tracking of the skeletal movement, it is seldom used as it is excruciating for the subject, takes a lot of time and could modify the subject's natural

motion (not directly because of the markers[28], but because of the local anesthesia[35]), while only bringing a little more accuracy in tracking [2].

This system does not solve the problems of occupied space, cost, time consumption and movement modification and only solves the precision problem.

2.2.5 Inertial Measurement Units

Nowadays, Inertial Measurement Units (IMUs) are constantly present in our daily life in our phones, drones, cars, etc. They consist of a combination of three captors: an accelerometer, a gyroscope and a magnetometer. Combined together and with the use of specific algorithms such as Gadgwick's filter, they allow the user to track the position of the IMU in a defined frame of reference. As Antonio I Cuesta-Vargas and his team showed in 2010, this technology can also be used in human motion analysis. The results are not comparable to standard motion tracking systems such as Vicon[7], but the errors can be acceptable depending on the application and on the part of the body that is tracked.

IMUs are subject to different systematic errors, namely bias, drift and scale factor, inherent to the different captors. There is a vast choice of different IMUs available on the market, with prices ranging from less than ten euros to thousands of euros per captor. Mattia Guidolin and his team compared the results of different IMUs in different price ranges and while the results are pretty similar in static conditions, when in dynamic conditions, the cheaper IMUs give worse results. It is important to note that the better results of more expensive IMUs are at least partially due to the proprietary software accompanying the sensors, an Xsens MTw Awinda, in this study.

Figure 2.6: Xsens MVN suit with 17 IMUs

The sensors' starting orientation needs to be known as well as the relative position between them. With the Xsens system, the calibration procedure consists of placing the subject in a

[2]It can still be used with less inconvenience in cadaverous subjects, even though the rigor mortis limits the skin and muscle displacements.

specific position, the T-pose shown in Figure 2.6. This calibration postulates a known and fixed distance between the different IMUs.

IMU systems have advantages; they can be low-cost and portable. They are not limited to a laboratory setting and can track the movement in an outdoor environment. The data computation is rapid and an instantaneous feedback can be obtained. Additionally, with IMUs, there is no risk of perturbation of the measurement due to the light or an obstacle in the field of view.

There are still some disadvantages, such as the risk of drifting after a certain utilization time, the sensibility to electromagnetic fields and, as always, the time-consuming task of setting the system up. Additionally, the cost of a high-end setup can still be high. It is essential to choose the system according to the needs of the problem studied; as Eline M. Nijmeijer and her team showed, the results are not always the same between a Vicon-like system and an Xsens-like system [34].

This system does not solve the problems of cost, imprecision, time consumption and movement modification and only solves the problem of occupied space.

2.2.6 Other methods

Other methods exist, such as dynamic radiography [29], but those are rarely used. In practice, different methods are often combined to corroborate and improve results. With the advent of modern computers and their always-growing capacity, new approaches were developed, more portable, faster and with less preparation. We will go through some of those methods in the next section.

2.3 Markerless methods

A critical advancement in recent years was the development of markerless methods to study the movement of the human body. This brings the huge advantage of being independent of the placement of markers on the subject and of the laboratory conditions while adding more difficulty in tracking this subject. However, today's computation power allows us to overcome this hurdle.

2.3.1 Human detection and tracking

The first step in developing markerless systems to study human movements is to be able to know when there is a human in the frame and be able to follow it. Indeed, in a system like Vicon, the only visible points in the image are the points of interest. Here, the background must be separated from the human while taking into account the occlusion and the self-occlusion in the foreground. Additionally, it is easy for humans to detect a human, but a program will have difficulties due to the light, clothes, morphology, etc.

Wu Yang and his team presented in 2005 a method to detect and track pedestrians [57]. In this method, the human is considered a deformable object, and they used a statistical model (using Boltzmann distribution and a Markov network) to predict the presence of a pedestrian. Their method works well for pedestrians in normal situations, and is even robust to partial obstruction during movement, but is only adapted to this situation. If the subject is crouching, he will not be detected. An illustration of a pedestrian tracking algorithm is shown in Figure 2.7

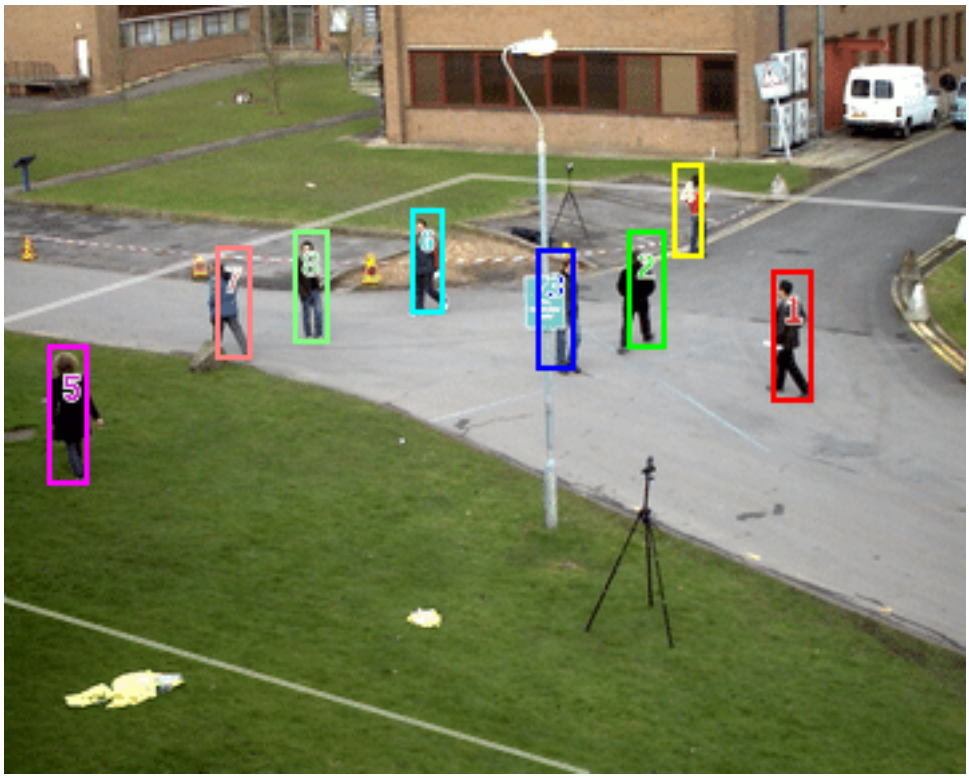


Figure 2.7: Results of a pedestrian tracking algorithm. [4]

Li Zhang and his team developed a solution with different models for different poses that can differentiate these poses and is thus not limited to standing people [59]. This method estimates articulation position to determine the global posture of the body. However, the articulation position was not their goal, for it still was the detection and tracking of people.

2.3.2 Pose estimation from 2D camera

Most modern pose estimation methods use only normal RGB cameras as input, either working on a single frame or a sequence of frames. In chapter 3 of their review from 2021, Jinbao Wang et al. [50] describe and compare no less than 41 different methods using a single 2D frame to estimate the 3D position of the articulations. Those 41 methods can be separated into three large categories, the methods estimating directly the 3D pose, methods estimating the pose in 2D first and then going to 3D and model-based methods.

Some direct methods work with a 3D probability heat map for each joint but are limited in resolution by the voxel size; smaller voxels give more precise results but increase the computational cost and memory consumption [39][27]. Other direct methods work as a regression problem and detect a root point, like the head, and use this point to estimate the other joints in a tree-like approach [25][60]. This method is limited in its generalization capacity as the bone length is fixed and does not adapt to the specific subject. Finally, some direct methods mix these two approaches [46].

Methods passing from 2D joint estimation to 3D joint estimation get the 2D estimation using different methods such as Neural Networks approaches [32] or other model-based approaches [45]. From this 2D information, some methods use additional depth information from other sources to reduce ambiguities between plausible poses[38][47][10]. Other methods use the tree-like structure of the joints to eliminate impossible poses knowing the normative distance between two defined

joints[33]. The 3D estimation propositions obtained are often re-projected in the 2D image to choose the most plausible estimation[48][10].

Regarding the model-based methods, the most commonly used model is the SMLP, standing for Skinned Multi-Person Linear Model[26]. The human model is deformed until its projection in 2D match the segmented human in the image. To accelerate the optimization process, a first estimation of the 2D joints positions can, as before, be used as a start of the fitting [5].

All those methods suffer common difficulties due to the frequent partial obstruction and the general ambiguity due to the passage from two dimensions to three. A common solution to those two problems is to use multiple views of the same scene. However, it is a difficult task to fuse the different points of view. A common approach is to work individually on every view and to fuse those results by taking the mean of the individual views[40][49][44]. Triangulation can also be used to reconstruct the 3D scene from different views[22][14]. The individual 2D estimations can also be used as a self-supervision tool for some models.

In chapter 4 of their review, Jinboa Wang and his team review another 16 methods using this time a sequence of frames instead of a single frame. The problem is relatively similar but with some advantages and some new challenges. The new challenges come from the always-changing background, the surroundings that are not fixed and easy to remove, the changing occluded parts of the tracked subject and the changing illumination. The advantages come mainly from the easier disambiguation of the poses inherited from the lack of depth information. Indeed, the pose between two consecutive frames can not be much different. Regression tasks and model fitting are much more manageable in image sequences than in single images.

2.3.3 Pose estimation from 3D camera

The previous subsection showed that depth information is rarely used outside of the disambiguation of poses obtained from RGB images. There are different methods working only with point clouds described in the review of Xu Tianxu and his team[58].

Some methods use human body templates to compare with the measured point clouds. These models can be separated into three big categories, the geometric models that divide the body in rough geometric shapes, the mathematical models that represent the body as probability distributions and the mesh models that represent the body like in video games as many small polygons that constitutes the surface of the body.

Other methods are feature-based and use the global and local shapes of the point cloud. Templates are used to find a starting point, such as the shoulder, and the other points are found via different algorithms knowing the normative distance between the different joints.

Finally, there are machine learning-based methods, the methods evolving the fastest in recent years. As stated previously, some methods use depth information as a disambiguation tool when working with 2D images and neural networks. Other approaches use depth data after transforming into a heat map or using it in a classification tree. Lastly, a method uses directly the point cloud in a neural network, combined with 2D estimations of the joints obtained by another algorithm.

All those markerless methods mainly solve the cost and time consumption problem and do require a dedicated room but do not solve the precision problem. Most of the markerless setups are not environment-dependent but those using depth data are still sensible to the environment.

2.4 The difference between Pose Estimation and Articulation Position Estimation

The finality of most approaches is not to determine the exact position of the articulation but to determine the body's global pose. The articulations are intermediate results used to animate a model in video games and animation movies or to determine an action done by the subject or a group of subjects in surveillance systems.

This implies that the precision of the articulations positions is not very high with some methods, which can be a problem in studying slight movement and small mobility problems. However, some methods are developed precisely for bio-mechanic analysis of the movement. In these applications, when working with a markerless method and point clouds, errors of between 6 and 10 centimeters are considered normal[58]. In recent commercial applications such as Theia Markerless[3], errors equivalent to the marker-based gold standard are promised, but this application is still in review [11][56][18][17][19][4].

2.5 Neural Network

An artificial neural network, often called a neural network, is an algorithm inspired by animal neural networks and part of the larger branch of machine learning. Those algorithms aim to achieve tasks that are too complex to be done by standard algorithms totally developed by a human programmer.

The neural network consists of neurons, called nodes, arranged in layers. There are an input layer, an output layer, and one or more hidden layers. The nodes of one layer are connected to one or many nodes of the next layer. An example illustration of this structure is shown in Figure 2.8. To each of those connections is assigned a weight, a parameter. Each node possesses an activation threshold that is passed when the sum of all the weighted inputs reaches a specific value. When activated, the node transmits a value to the next layer. Otherwise, it stays off and does not transmit anything.

The neural network is trained with a set of data that is given as input. The typical input data will be an image, otherwise represented as a list of numbers. The different parameters are at first set randomly. Every layer becomes the input of the following layer until the output layer. For each input, a known output is expected during the training process, and an unknown output is predicted when using the trained neural network. An error function is computed to measure the error amplitude between the expected output and the obtained output. The weights are then updated proportionally to this error, with an additional random modification to avoid falling into a local minimum. To ease the convergence, the weights are not updated for every input but for every input batch.

2.5.1 Hyper-parameters

Hyper-parameters are parameters of a neural network that are set before the training process and stay constant or semi-constant during the training process. Those hyper-parameters have no mandatory value but have a non-negligible impact on the quality of the training. Here are some of the important hyper-parameters:

[3] https://www.theiamarkerless.ca/

[4] Note that 3 of those 5 papers were written by the same research team and co-signed by the CEO of Theia

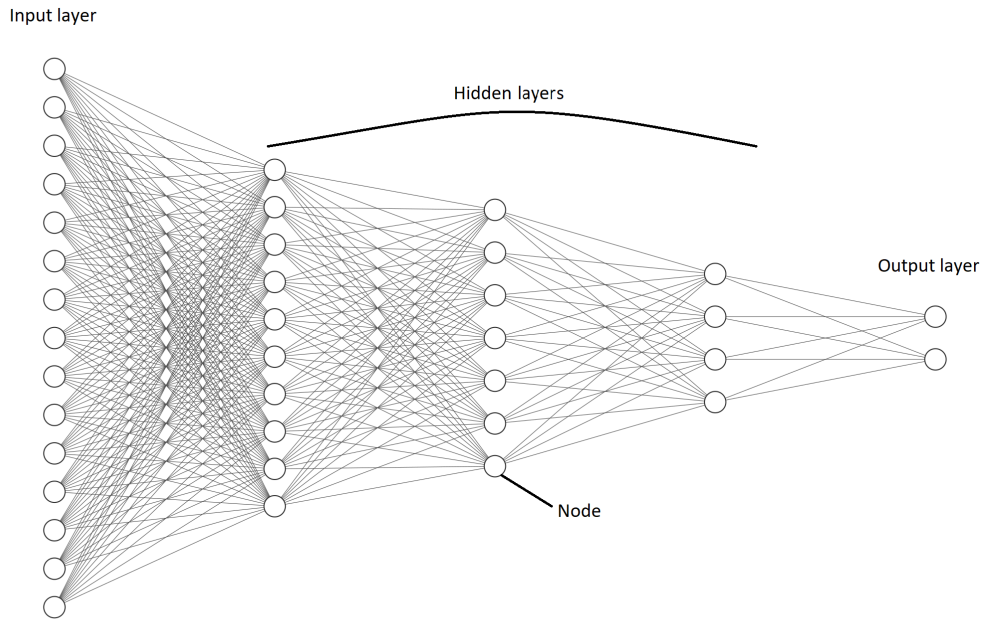


Figure 2.8: Example of a neural network structure.

- Loss function: this function measures the difference between the desired and resulting outputs. Choosing a loss function adapted to the problem that the model has to solve is crucial. The loss function for a binary classification will not be the same as that of a regression, let's say finding the center of a square placed randomly in an image. For example, in the case of binary classification, a cross-entropy loss will be chosen, while in regression, the distance between the found and wanted points can be used.

- Optimizer: this is the function used to calculate the new parameters of the neural networks. The goal of the optimizer is to minimize the loss based on the gradient of the loss and the learning rate. Many optimizers exist with specific update rules, but they will not be reviewed here. The Adam optimizer, the one used in this work, is described in the subsection 3.1.4

- Learning rate: this is a hyper-parameter that will define how much the parameters of the neural networks change every update. A higher learning rate will lead to a faster convergence but also to instabilities and even sometimes a divergence.

- Number of epochs: an epoch is a single complete iteration through the entire training dataset. We generally use multiple epochs for our training. A model trained on too few epochs will not have assimilated all the characteristics of our dataset, but with too many epochs, it will over-fit and not be able to adapt to new unseen data.

- Batch size: the parameters of a model are not updated after every item but after every batch of items. A large batch size has the advantage of reducing the load on the computer, but the model will have more difficulties generalizing [20] and may also have memory issues.

Chapter 3

Methodology

Chapter summary: This chapter is separated into three parts: the first part explains the choice and structure of the Pointnet neural network we used, the second part presents the dataset used and the different preprocessing steps as well as the data management strategies and the last part overviews the material used.

3.1 Neural Network

3.1.1 Why this method?

We chose to use a machine learning approach as it proved really efficient when working on our type of problem in recent years[58], while more traditional computing methods have difficulties converging and giving good results on such complex tasks. We chose to work directly with the depth data as a point cloud formed by the fusion of multiple cameras, a method not yet studied as far as we know. This will allow us to work with raw data not altered by different transformations when passing in two dimensions.

We chose to use the PointNet network structure as it is the first neural network to directly work with point clouds as input data[42]. PointNet was developed firstly to tackle classification and segmentation tasks on point clouds, not for regression tasks like the one we want to implement. Still, the network can be lightly modified to be adapted to our needs.

3.1.2 Network structure

The PointNet model was developed by Charles Ruizhongtai Qi and his team in 2016 [42]. The global structure is schematized in Figure 3.1 and consists of a series of convolutions, batch normalization and rectified linear unit (ReLu). In our implementation, we limit ourselves to the classification network. The activation function is linear instead of the soft-max used classically when using PointNet for classification. Our final implementation structure is detailed in Table 3.1, considering a 5000 points point cloud input. We decided not to add the two small T-Net that are present in the original version to ease the implementation and to reduce the computing time of our implementation. The role of these T-Net is to make the whole network robust to translations and rotations of the input point cloud, we will thus not use translations and rotations in our data augmentation process.

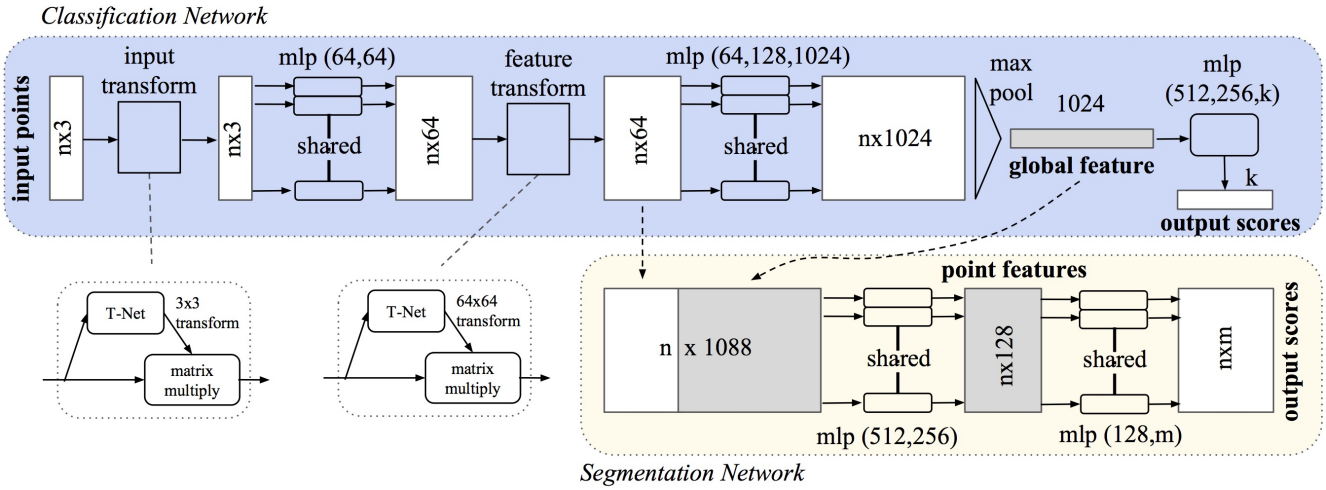


Figure 3.1: PointNet network Architecture [42]

Layer (type)	Output Shape	Param #
Conv1d-1	[-1, 64, 5000]	256
BatchNorm1d-2	[-1, 64, 5000]	128
ReLU-3	[-1, 64, 5000]	0
Conv1d-4	[-1, 64, 5000]	4,160
BatchNorm1d-5	[-1, 64, 5000]	128
ReLU-6	[-1, 64, 5000]	0
Conv1d-7	[-1, 64, 5000]	4,160
BatchNorm1d-8	[-1, 64, 5000]	128
ReLU-9	[-1, 64, 5000]	0
Conv1d-10	[-1, 128, 5000]	8,320
BatchNorm1d-11	[-1, 128, 5000]	256
ReLU-12	[-1, 128, 5000]	0
Conv1d-13	[-1, 1024, 5000]	132,096
BatchNorm1d-14	[-1, 1024, 5000]	2,048
ReLU-15	[-1, 1024, 5000]	0
Linear-16	[-1, 512]	524,800
BatchNorm1d-17	[-1, 512]	1,024
ReLU-18	[-1, 512]	0
Linear-19	[-1, 256]	131,328
BatchNorm1d-20	[-1, 256]	512
ReLU-21	[-1, 256]	0
Linear-22	[-1, 57]	14,649

Total params: 823,993
Trainable params: 823,993
Non-trainable params: 0

Table 3.1: Structure of the PointNet neural network used with a 5000 points point cloud

3.1.3 Loss function

In our problem, we want to estimate the position of 19 points in space, corresponding to the joints and points of interest described in section 3.4. We have the correct coordinates of these points and our network outputs 19 points coordinates. The most straightforward loss function would be to take the distance between the points, but this loss would vary in sign, which is not what we want as we will minimize this function but want it to reach zero. Two options to solve this problem are offered: the absolute value of the distance or the squared value. We chose to take the squared value to get a more significant impact on the loss from the more significant errors in the estimation. This is the classical loss in regression problems, Mean Squared Error (MSE)

3.1.4 Optimizer

We chose to work with the optimizer Adam, a widely used optimizer [21]. One of its particularities is that it adapts the learning rate for every parameter based on the gradient of this parameter to ease a faster convergence.

3.2 Dataset

3.2.1 Dataset selection

The choice of the dataset is one of the most important steps when working with a neural network, if not the most important one [15] (Garbage in, Garbage out [1]). As we weren't able to create from scratch a dataset in the laboratory due to time constraints, we had to find an existing one.

We looked for a dataset with certain criteria:

- Dataset of human body capture
- Multiple cameras around the subject
- Time of Flight depth data from the multiple points of view
- Large amount of data
- Ground truth for the joints positions
- Freely available
- Easy to manipulate

Different options were available to us, with always some advantages and some disadvantages. A comparison of the characteristics of four datasets is shown in Table 3.2. Those datasets are the Visual Computing Laboratory dataset (VCL)[1], the Smart Walker Dataset (SWD)[37], the Human3.6M dataset[13] and the CMU panoptic dataset[16].

[1]Popular concept in computer science stating that nonsense input data will give nonsense output. https://en.wikipedia.org/wiki/Garbage_in,_garbage_out

	VCL	SWD	Human3.6M	CMU Panoptic
Human body	+	+	+	+
Cameras around the subject	+	-	0	++
Multiple ToF	+	0	-	+
Large amount of data	0	+	++	+
Ground Truth	0	+	+	-
Availability	+	+	-	+
Ease of use	–	+	0	0
score	3	4	2	5

-:Not good; 0:Neutral; +:Good; ++:Very good

Table 3.2: Comparison of different datasets

The Visual Computing Laboratory (VCL) developed many datasets, such as the "Datasets of multiple Kinect2 RGB-D streams and skeleton tracking"[1], but while some of those are datasets from multiple ToF cameras placed around the subject, the sequences are too short, with almost no ground truth. Additionally, many download links are unavailable, and the tools proposed to read and use the homemade file types do not work on the last versions of Windows. While promising, this dataset is thus not usable for us.

The Smart Walker dataset [37] is an extensive dataset of more than 160 thousand frames, with a ground truth based on Inertial Measurement Units and a toolkit proposed to use the data that is readily available on GitHub and Phisionet. The significant disadvantage of the dataset is that only two ToF cameras are placed in front of the subject and do not capture the same scene. This is too different from our laboratory setup, which leads to us leaving this dataset aside.

The Human3.6M dataset [13] is a reference in terms of the human pose dataset with, as its name states, 3.6 million poses available. A ground truth is available, based on a Vicon-like system, but only one ToF camera is available. The dataset itself is blocked behind a connection page, and to get access, the page admins have to grant it to you, but the project seems to be paused since 2020. Once again, this dataset does not fit our needs

The CMU panoptic dataset consists of hundreds of cameras placed around a subject, among which 10 ToF cameras. The data is readily available, with tools to download and work with the raw data. The big default of this dataset is the lack of ground truth available for the skeleton position. The skeletons were not obtained through a Vicon or an IMU system but via an algorithm developed by the teams at the CMU. However, the results obtained, described in subsection 3.2.5, were deemed sufficient for this thesis.

3.2.2 CMU Panoptic

The CMU Panoptic Dataset was developed by a team at Carnegie-Mellon University, composed of Hanbyul Joo, Tomas Simon, Donglai Xiang, Yaadhav Raaj, Professor Yaser Sheikh and other collaborators between 2016 and 2019.

It consists of 521 synchronized videos, more precisely 480 VGA cameras (640x480 pixels (p), 25 fps (Frames per second)), 31 HD cameras (1920x1080p, 30 fps) and 10 RGB-D cameras (RGB: 1920x1080p, 30 fps; D: 512x424p, 30 fps). Those cameras are evenly distributed on a 5.49-meter-wide geodesic sphere. There is a total of 132 sequences with a total of many hours of synchronized

video. Unfortunately, not all sequences were recorded with all the cameras. The structure of the Panoptic studio is shown in Figure 3.2

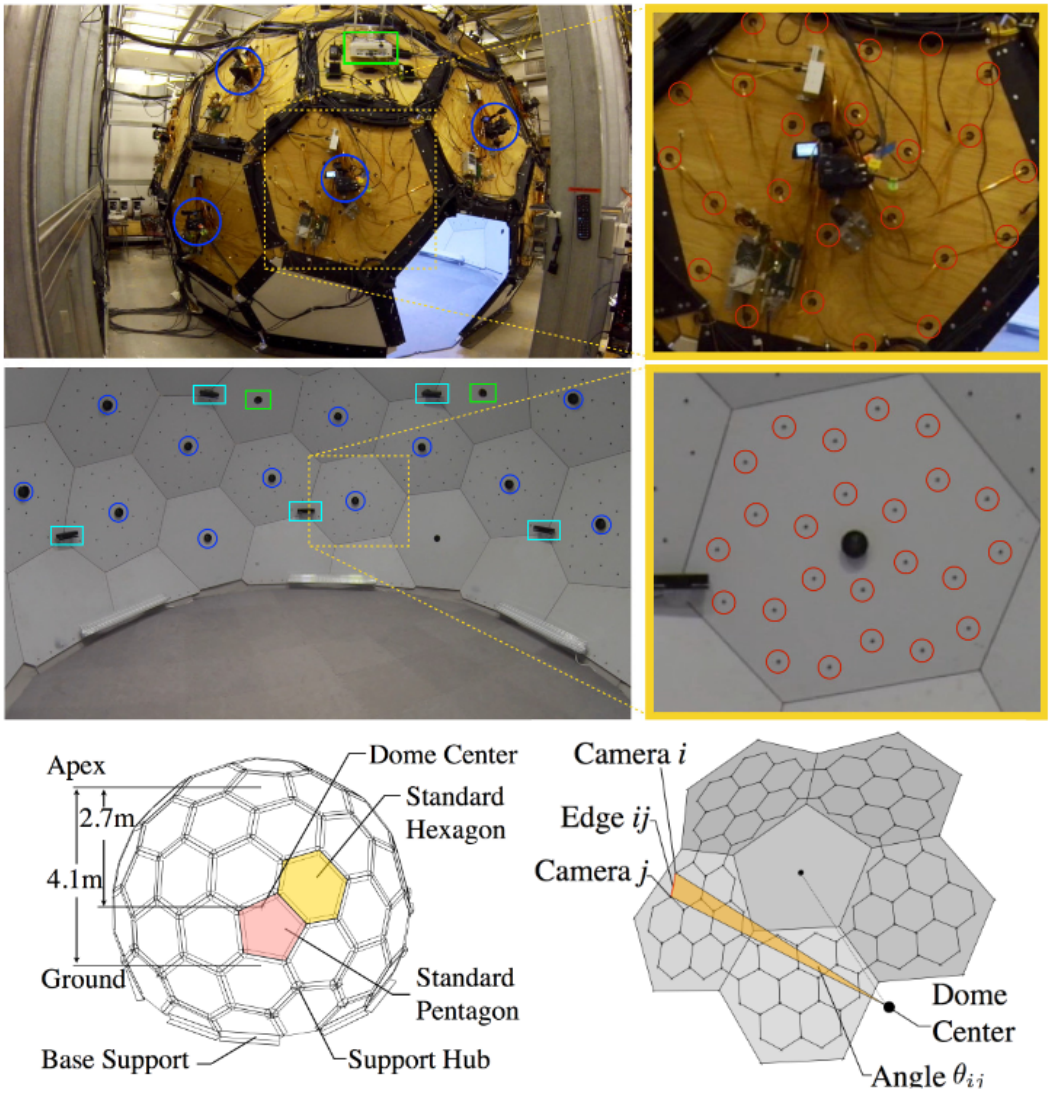


Figure 3.2: Structure of the panoptic studio. VGA cameras are circled in red, HD cameras in dark blue and RGB-D in light blue. The green rectangles are projectors [16].

3.2.3 Dataset structure

The dataset can be downloaded easily by sequence and with the modalities wanted. For each sequence, we can download the body skeleton information, the hand skeleton information, the face feature-point information and the synchronization data of all the different types of cameras. Obviously, we can download the video feed itself, with all or a subset of each type of camera. For the RGB-D cameras, depth data is also available.

The sequences are classified into different movement types or situations:

- Range of movement (9 sequences)
- Haggling (34 sequences)
- Ultimatum (6 sequences)
- Mafia (7 sequences)
- Dance (14 sequences)
- Musical instruments (23 sequences)

- Toddler (17 sequences)
- Special event (22 sequences)

Haggling, Ultimatum and Mafia are the names of the game played by the subjects in the videos. In this thesis, we will focus on single adult subjects moving in the field of view and will thus focus on the first set of sequences, Range of movement.

3.2.4 Data used

There are 9 "range of movement" sequences. We will use only one of them to train our networks. The one we will use is "171204_pose1", a 17 minutes and 30 seconds sequence with six subjects repeating a set of movements. They are of different gender, height, morphology and skin tone. In this sequence, the subjects follow a set of movements while standing at the center of the room. The movements include arms, legs and shoulder raises, hip and neck rotation, jumps and crouches.

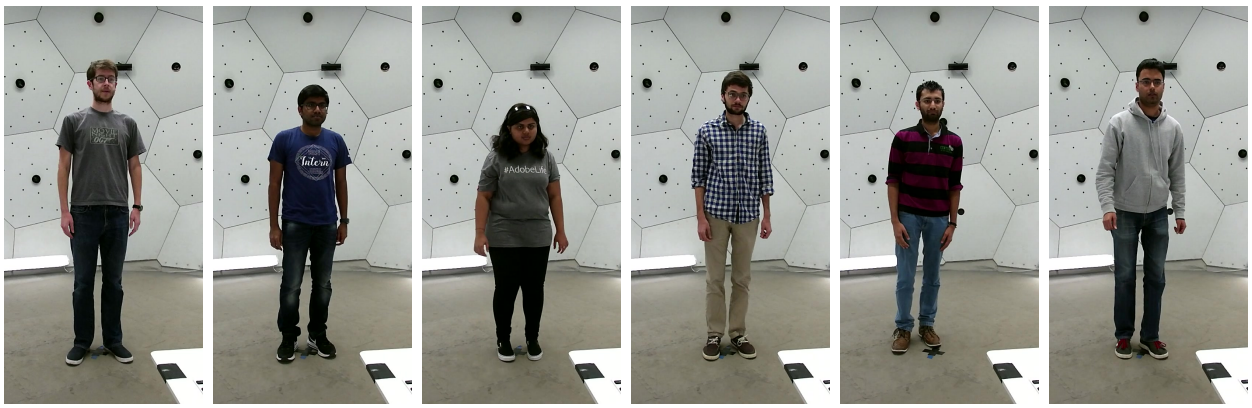


Figure 3.3: The six subjects present in our training sequence.

Our first approach was to split the "171204_pose1" sequence into two subsets, a subset for the training and a subset to test the results. To keep a maximum diversity during the training process, we chose to do this split with around 20% of the frames in the test subset, chosen in a non-continuous way, i.e., one frame every five frames. The problem with this approach was that with our 30 frames per second, the difference between two consecutive frames is almost negligible due to the low speed of human movement. The test models trained with this separation between training and test subset gave results too good to be true, far surpassing the state of the art. This approach was thus discarded.

To test the results, we will instead use another subject from another sequence, "171204_pose2", who's not in the first sequence and performs a similar set of movements. This test sequence is 2 minutes and 1 second long.

3.2.5 Target data

The Kinects used to capture the depth data in this dataset come with an algorithm to estimate the skeleton position. This algorithm, while giving quite accurate results, is destined to work on single cameras. Due to the difficulty of fusing the results of the different cameras, the team chose to develop a new algorithm based mainly on the huge amount of points of view available.

In each VGA camera view, they generate a 2D pose estimation using the pose detector of Wei and al. [51]. The result in each frame is approximate, especially due to the partial self-occlusion

of the subject. The 2D joints obtained are projected in 3D with a presence probability in every voxel of the space, and then the more likely 3D propositions are kept for each joint.

The next step is to generate parts by linking the different joints together. The parts that fit a maximum of views when de-projected in 2D are kept. We now have a skeleton proposition in 3D. They worked on scenes with sometimes more than one subject. The skeletons are tracked through time by looking at the skeleton in the precedent frame with the smallest distance between the head joint.

This method gives good results for scenes with one or few subjects. Still, it starts to have difficulties coping with many subjects simultaneously, mainly because of the large and frequent obstruction in many views for a specific subject. A refined method was developed to solve this issue with temporal information and spatial tracking of the subjects [16].

The complete method gives accurate results with 99.32% of the joints correctly estimated and 93.55% of skeletons with all joints correctly estimated. Joints are considered correctly estimated if they are less than 5 centimeters apart from manually annotated joints.

3.2.6 Data preprocessing

We need to prepare the data before using it. Many preprocessing steps were implemented to get the cleanest inputs in our networks and are illustrated in Figure 3.4.

1. Transforming video to images: Videos are impractical to work with, so we first need to transform the video into a series of images. To do so, we used FFmpeg tools [2] to decompress the videos.

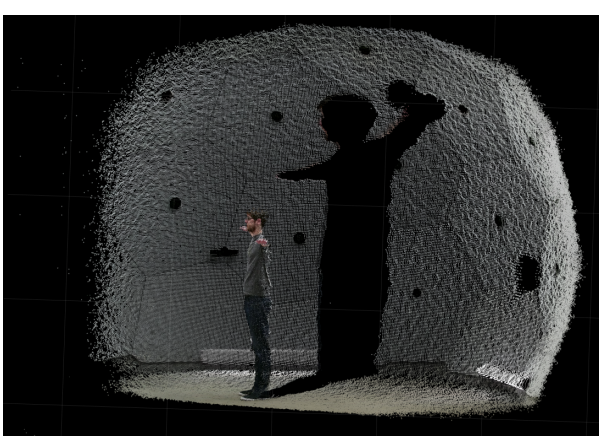
2. Generating point clouds: We generate the point clouds using the image from the last step and the depth data, obtaining a set of points in a 3D space. The point clouds are saved in a ply file, with the three spatial coordinates and the three color values in RGB for every saved point.

3. Placing the point clouds in the same reference frame: To be able to fuse our cloud, we need to put them in the same reference frame. Knowing the positions of cameras, it is easily done with some rotations.

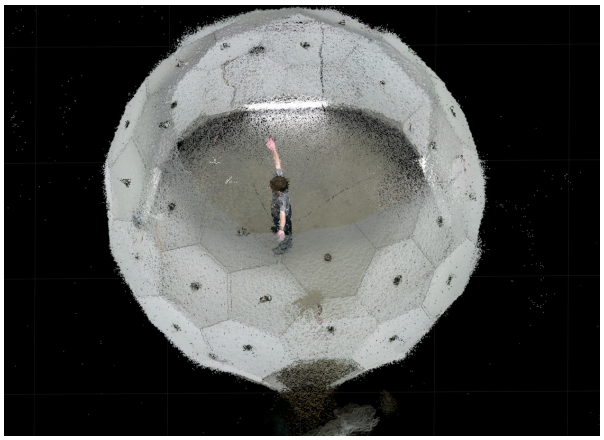
4. Walls and floor removal: We only want to study the subject. We can thus remove the walls and floor points. To remove the floor, we remove the lowest centimeters in the cloud. To remove the wall, we remove points further than two meters from the center of the geodesic dome. This and the two precedent points of the preprocessing were made in MatLab using the panoptic toolbox[3].

5. Point cloud cleaning: There is some random noise in the point cloud. To remove most of it, we only keep the points that are in a dense region of the point cloud.

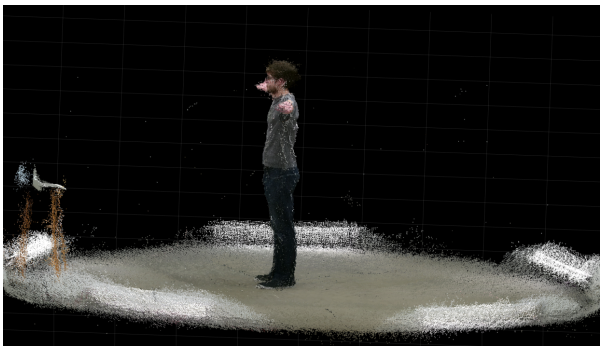
[2] FFmpeg is a cross-platform solution to record, convert and stream audio and video (https://www.ffmpeg.org/)
[3] Available on GitHub https://github.com/CMU-Perceptual-Computing-Lab/panoptic-toolbox



(a) Point Cloud of one camera



(b) Fused Point Cloud of ten cameras



(c) Point cloud without walls

(d) Point Cloud without floor and cleaned

Figure 3.4: The different steps of the preprocessing

We now have a clean colored point cloud of our subject for every frame and every camera, stored as a "ply" file. We chose not to directly fuse the point clouds to limit the already ample preprocessing time and to be able to do any needed combination of cameras later on. A new challenge appears as every frame size is between 2 and 5 megabytes, a solution must be found as it will not be efficient to work with more than 150Go of data.

3.2.7 Frame selection

Frames lacking a ground truth or with a not dense enough point cloud were filtered out. We save the indexes of the correct frames in a CSV file. The threshold for the point cloud density is arbitrarily fixed at 2000 points to guarantee a certain density. The different cameras have between 14417 and 25997 valid frames with a mean point cloud size of 3428 points. The two cameras with less than 21000 valid frames are the lateral ones, cameras 3 and 6.

3.2.8 Data Management

Working with tens of gigabytes of data in a neural network is problematic as it will take much too long to load and compute those data. In order to reduce the size of the data, multiple steps were taken as follows:

1. Remove color data: we will not use the color data of the cameras.

2. Store the points in a numpy array and save them in an NPY file: numpy arrays are a very efficient way to store data.

3. Store every frame in the same file: this step does not improve the global size of the data but will limit the number of memory accesses in the hard drive. However, loading tens of gigabytes in the RAM is not possible. To avoid this problem, we chose to work with the hdf5 file format that only loads the pointers to the different frames in the RAM

4. Keep the data near the computer: the original data was too voluminous to stay in the local hard drive of the computer, everything was then stored on a distant data server. This added to the loading time with an additional network connection step at every memory access. Now that the data is reduced to a few tens of gigabytes, we can store it in the local hard drive.

The hdf5 files are structured like a classical folders and files system. The training file has ten entries, one for each camera and, within each entry, thousands of frames. Each frame has then thousands of points. We have 4 hdf5 files, one skeleton and one point cloud file for the training and testing of our model.

The final data preprocessing and the data management operations took 317 hours of computation time.

3.2.9 Data augmentation

Every frame consists of between a few thousand points and over 25000 thousand points. We chose to train our models with a 5000-point cloud input. Instead of preprocessing this step, we select a random set of those points on each frame for each epoch of training. When using multiple cameras, we first fuse the cameras and then take the random sample, and if the amount of points is too low for the set number of points desired, we duplicate all the points and then select the random sample. If, for a certain frame, a camera has no valid data, we keep the frame with the information coming from the other cameras.

3.3 Hardware

The first tests and part of the preprocessing were done on a laptop that has for CPU an Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz, 8 Go of RAM and no usable GPU.

The rest of the preprocessing and the training and the computation of the results were done on one of the computers of the LISA laboratory. It has for CPU an Intel(R) Core(TM) i7-7800X CPU @ 3.50GHz, 32Go of RAM and a NVIDIA GeForce GTX 1080Ti for graphics card. The training was accelerated using the GPU.

The data was stored first in a NAS system and later on an SSD hard drive on the computer.

3.4 Skeleton model

We will represent the skeleton with 19 joints and points of interest. The joints are defined following the current key point order, shown in Figure 3.5: 0: Neck, 1: Nose, 2: Body Center (center of hips), 3: Left Shoulder, 4: Left Elbow, 5: Left Wrist, 6: Left Hip, 7: Left Knee, 8: Left

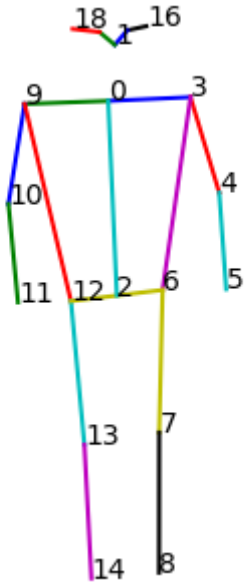


Figure 3.5: Skeleton model used in this work. The two eyes are not labeled on the image for display clarity.

Ankle, 9: Right Shoulder, 10: Right Elbow, 11: Right Wrist, 12: Right Hip, 13: Right Knee, 14: Right Ankle, 15: Left Eye, 16: Left Ear, 17: Right Eye, 18: Right Ear.

3.5 Implementation

Our code was developed in Python using the Pytorch machine-learning framework. The computations were accelerated using numpy and pandas. Different additional libraries were used for specific tasks. The code of this project can be found on the following GitHub page:
https://github.com/BenjiPoly/MEMO-H500_JointEstimation.git

Different aspects of our code use pseudo-random computation. To ensure repeatability, the seeds are all fixed to a common number.

Chapter 4

Results

4.1 Proof of concept

Now that we have a model and a dataset, we first need to assess our capacity to predict the position of the major joints of the human body with our data. For the first test, we used a very straightforward approach with the following hyper-parameters:

- Batch-size = 80; we update the weight of the parameters after processing 80 frames
- Epoch = 1; we train the model on all the frames one time
- Optimiser = Adam;
- Learning rate = 0,001; we chose a starting learning rate relatively high to converge rapidly
- Loss function = MSE;

This model was really too simple and its capacity to learn in one epoch was too limited. The results appear relatively good for the most common poses with a mean error for the estimated joints of 14.35 centimeters, but there is no learning for the less common poses, as we can see in Figure 4.1. For a crouching pose, the model tries to fit a standing skeleton in the point cloud, giving a small skeleton and a mean error for the estimated joints of 28.76 centimeters.

This first attempt being too simple, we need to try one a bit more complex before going further in our study. We chose to train a new model longer and slower with the following hyper-parameters:

- Batch-size = 80;
- Epoch = 5;
- Optimiser = Adam;
- Learning rate = 0,0001; we chose a starting learning rate relatively low to converge smoothly.
- Loss function = MSE:

As we can see in Figure 4.2, this model gives much more acceptable results for the crouching pose with a mean error of 19.41 centimeters while still being far from exact. The results for the standing pose are almost perfect with this model, with a mean error of 5.69 centimeters.

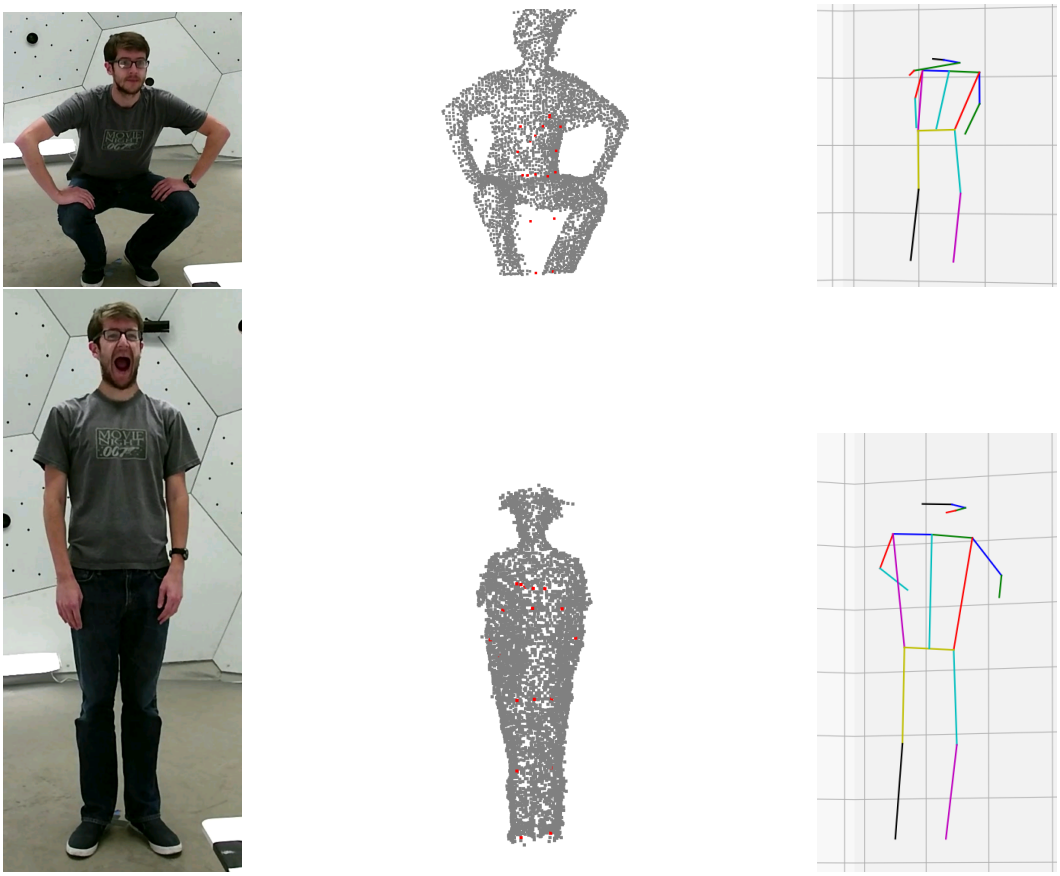


Figure 4.1: Results of the joint estimation with one epoch of training

Figure 4.2: Results of the joint estimation with five epochs of training

Those two tests were done using the CPU (Central processing unit) of a laptop. The training time being too long, the rest of the training was done on another computer equipped with an NVidia GPU, allowing us to use Cuda optimizations.

4.2 Experiments

In this section, we will speak of the cameras using their label, from 1 to 10. A neural network model trained with as input the images from camera 2 will be named "model 2". When a combination of cameras is used, the name will be the concatenation of the different camera labels in ascending order. For example, a neural network model trained with the combined images of camera 2, camera 5, camera 6 and camera 9 will be named "model 2569" We schematize the placement and number of the different cameras in Figure 4.3.

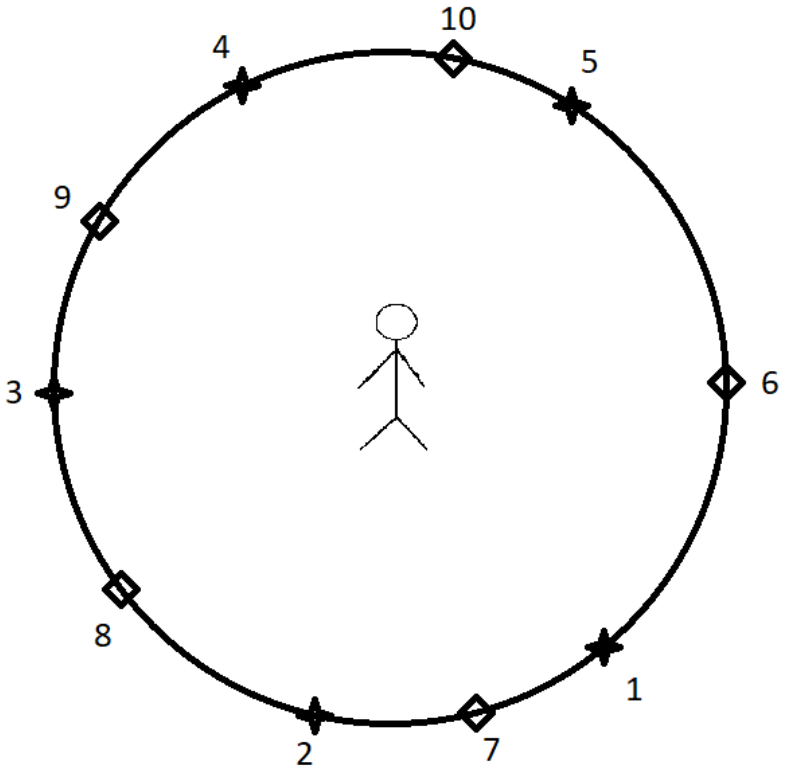


Figure 4.3: Cameras placement and orientation around the room. Star cameras are placed around a meter high and diamond cameras are placed around two meters high. The subject looks between cameras 2 and 7

4.2.1 Number and position of cameras during training

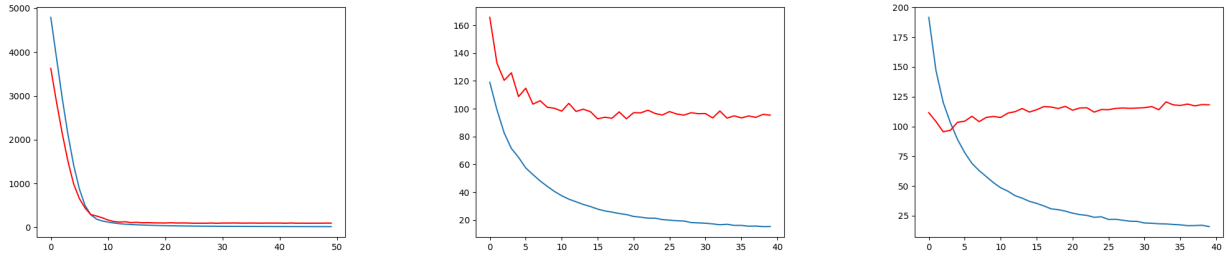
To study the influence of the camera position and the number of cameras, we will train different networks with the different cameras alone and then with varying combinations of a growing number of cameras.

We chose two types of setups for the multiple camera models. We either picked cameras placed near each other or evenly distributed around the room.

The combinations we chose are 2-5, 2-7, 4-7, 5-10 and 1-2 for the duos, 2-5-9, 1-2-7 and 1-2-3 for the trios and 2-5-6-9 and 1-2-7-8 for the quatuors. We will thus work with 20 different models with the same hyper-parameters. Those hyper-parameters were set as follows:

- Batch-size = 40; we update the parameters of the nodes after processing 40 frames
- Epoch = 50; we train the model on all the frames 50 times
- Optimiser = Adam;
- Learning rate = 0,0001; we chose a starting learning rate relatively low to limit the starting over-fitting.
- Loss function = MSE;

After 100.3 hours of training, we now have 20 models ready to be tested. After 20 epochs, most models converged (Figure 4.4a). For some models like the model 2569, there was still a tendency for an improvement of the test loss (Figure 4.4b), but for some, an overfitting appeared (Figure 4.4c). We decided to still compare all the models at the same training epoch. The first step is to get the results for each model for each frame of the test sequence, with each camera and some combination of cameras. We chose to test the models for each individual camera and for each combination of up to four cameras. We thus have 385 camera combinations to test 20 times.



(a) Training and test loss on 50 epochs. Model 2569

(b) Truncated training and test loss after epoch 10. Model 2569

(c) Truncated training and test loss after epoch 10. Model 8

Figure 4.4: Training (blue) and test (red) losses

After 429.4 additional hours, we have the results of all those tests, but we need to define a way to compare and classify the different obtained models.

Three criteria are used:

- Mean
- Standard deviation
- Quality

This last criterion is often used as a comparison in the literature [58] and consists of the percentage of frames with an error below a threshold of 10 centimeters. The three criteria are calculated individually for every articulation in every frame for every camera combination. A global mean of the criteria on all articulation is then calculated and used to classify the models.

In Table 4.1, every numbered row is the result of one model, and every numbered column is for a different joint. The joints are defined following the model described in section 3.4. The tables in Table 4.1 are colored with a gradient from green, the best results, to red, the worst results and the models are sorted from the best to the worst mean score for all the joints in each table.

Combining the results of those three criteria by summing the individual ranks of the models in the three criteria, we arrive at a global classification of the models that is presented in Table 4.2.

Model	2569	127	6	12	9	259	1278	4	123	7
Global Score	3	6	10	21	21	22	23	29	31	32

25	2	510	10	24	3	8	5	47	1
35	35	39	39	43	43	47	48	49	54

Table 4.2: Global Ranking of the 20 models

Mean	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Mean
2569	6,79	8,42	5,10	7,45	9,82	15,49	5,43	6,71	7,23	7,40	11,77	17,65	6,08	5,81	6,23	10,04	8,80	8,06	8,95	8,59
127	6,49	7,30	7,41	6,92	13,67	22,97	7,42	6,81	5,79	8,61	14,50	17,80	9,16	9,52	9,60	8,98	8,05	10,55	10,93	10,13
6	7,93	10,40	5,98	8,21	11,28	20,92	5,97	7,21	6,18	9,02	14,39	20,06	6,88	8,97	7,62	11,46	10,96	10,44	10,51	10,23
12	8,82	9,32	7,66	8,10	10,90	16,13	6,97	7,75	6,83	11,05	14,70	19,09	10,02	10,41	12,37	12,56	13,35	13,49	13,80	11,23
9	9,77	11,89	8,51	11,16	11,56	19,27	9,15	11,68	10,12	9,74	10,70	16,92	9,02	11,16	11,21	12,66	12,20	11,33	11,84	11,57
1278	8,19	9,37	8,41	8,32	12,60	19,01	8,26	8,17	8,16	9,19	14,08	19,39	9,90	11,83	13,30	12,62	13,60	13,11	13,15	11,61
259	8,92	10,11	7,77	9,83	14,26	23,41	8,61	9,80	10,58	9,42	14,01	22,51	9,01	9,86	9,15	13,49	13,85	12,21	13,62	12,13
7	9,13	9,34	11,09	8,95	16,03	26,28	10,76	9,17	7,32	10,20	14,93	24,42	11,47	10,80	11,40	10,72	10,90	10,33	10,63	12,31
123	11,50	13,09	9,10	11,08	11,43	19,57	8,71	8,87	6,76	12,25	14,91	23,52	10,58	13,44	12,87	14,47	14,26	11,42	11,89	12,62
4	10,12	11,81	10,26	10,78	15,62	24,23	10,36	9,90	8,87	10,77	15,11	20,47	11,23	13,15	12,26	12,72	12,22	11,60	12,14	12,82
25	11,45	13,13	9,58	11,38	13,24	18,83	9,45	8,68	8,79	12,82	16,02	21,74	11,67	10,15	10,37	13,75	14,20	15,00	14,67	12,89
10	10,51	12,36	8,75	9,79	12,84	22,25	8,69	17,50	17,89	11,40	16,04	21,91	9,50	8,50	6,45	13,04	11,95	12,87	12,89	12,90
2	11,40	12,15	10,91	10,48	14,16	19,30	10,75	10,78	9,24	12,98	15,54	17,07	12,51	12,26	11,68	14,24	15,74	14,72	14,59	13,18
24	11,00	11,74	11,63	10,32	12,15	21,19	11,44	11,64	9,94	12,42	14,78	22,72	13,12	12,94	13,08	14,81	15,64	12,50	12,90	13,47
510	11,04	11,45	9,66	10,24	14,38	22,54	9,91	11,38	10,72	13,24	20,73	26,12	12,18	12,07	11,94	12,62	13,10	13,00	12,65	13,63
5	12,19	14,01	8,57	11,44	17,94	25,19	8,09	8,07	9,64	15,27	20,94	24,77	11,57	9,78	10,35	14,94	15,09	15,17	16,32	14,18
8	11,65	13,76	12,04	11,59	15,43	25,61	12,36	12,39	11,39	12,73	16,68	24,79	12,59	11,16	12,93	13,37	13,40	14,22	14,12	14,33
47	13,90	15,92	12,50	13,07	12,39	18,35	12,33	12,12	11,91	15,44	16,29	21,07	14,32	14,03	13,83	17,25	17,21	15,76	15,29	14,89
1	14,90	16,95	10,00	13,15	16,03	23,94	10,57	14,78	13,35	17,59	18,99	26,41	11,75	11,73	11,95	18,67	19,43	15,93	16,71	15,94
3	17,10	17,98	12,29	16,74	16,86	22,74	11,93	12,56	12,15	17,81	15,39	16,12	13,33	13,38	12,28	19,98	20,63	18,81	18,79	16,15
Joint Mean	10,64	12,03	9,36	10,45	13,63	21,36	9,36	10,30	9,64	11,97	15,52	21,23	10,79	11,05	11,04	13,62	13,73	13,03	13,32	

SD	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Mean
2569	2,57	3,05	2,88	3,08	5,07	7,68	6,03	4,31	4,43	2,86	5,11	7,91	5,48	4,56	4,41	6,21	6,78	7,35	9,52	5,23
127	2,96	3,58	3,28	3,45	6,38	11,13	6,19	5,11	4,87	3,13	6,29	7,86	5,82	5,69	5,59	7,41	8,42	6,17	7,46	5,83
3	4,10	4,55	3,52	4,51	7,53	10,16	5,17	4,94	6,33	4,19	5,04	7,61	4,96	5,97	6,45	7,23	8,17	5,75	6,44	5,93
6	2,53	3,46	3,81	2,81	5,69	9,90	6,21	4,91	5,71	2,89	6,74	10,75	5,80	5,57	5,74	9,98	11,67	5,72	6,15	6,11
2	3,60	3,68	4,00	3,24	5,73	8,82	5,87	5,18	5,29	4,88	7,72	9,56	6,34	5,85	7,38	9,18	10,66	6,69	6,78	6,34
12	3,68	4,16	4,31	3,56	5,08	7,32	6,53	5,72	5,42	4,91	7,46	10,29	6,73	7,20	7,26	8,90	9,90	6,87	7,56	6,47
1278	3,80	4,57	3,77	3,61	6,28	9,70	6,16	5,37	5,04	4,58	6,04	8,85	6,54	7,54	7,76	9,24	9,95	8,51	9,39	6,67
9	4,27	5,07	4,79	4,46	6,41	8,81	6,96	6,36	6,87	4,71	7,14	9,36	6,96	7,18	7,30	11,35	12,43	6,95	7,06	7,08
259	4,83	5,46	4,61	5,07	7,32	11,43	7,09	6,16	6,24	5,08	7,42	11,41	6,79	6,11	6,24	7,89	7,97	8,48	9,95	7,13
4	4,86	5,15	4,79	4,31	6,44	11,08	7,04	5,39	5,63	5,59	6,86	11,09	7,35	6,76	7,88	8,90	9,54	7,99	9,90	7,19
8	4,12	5,72	4,17	4,71	8,37	13,25	6,44	5,36	6,10	4,47	7,54	9,87	6,87	6,77	6,56	10,35	11,42	7,44	8,11	7,24
7	4,38	4,94	3,90	4,75	7,33	12,68	5,78	6,40	6,74	4,47	8,16	13,52	5,56	6,49	6,87	10,66	11,66	6,56	6,80	7,24
10	4,43	4,56	4,03	4,42	5,95	12,10	6,64	7,75	7,95	5,17	7,32	12,53	6,43	5,48	5,72	10,47	11,69	7,58	8,14	7,28
24	4,89	5,97	3,93	4,95	5,95	10,51	6,56	5,46	5,36	5,61	7,17	10,74	6,77	7,07	7,53	11,94	12,62	8,47	9,01	7,39
123	5,43	6,30	5,28	4,86	5,89	9,11	7,06	6,33	6,01	6,36	7,12	9,48	7,54	8,99	8,76	9,08	9,41	8,01	9,53	7,40
5	4,80	5,70	3,68	4,97	8,65	13,67	7,16	5,66	5,27	5,81	9,50	13,29	7,17	6,35	6,49	10,04	10,93	7,56	8,21	7,63
1	5,68	7,11	4,11	4,68	7,40	11,61	6,55	6,24	5,85	7,00	8,30	11,53	7,00	7,69	8,35	11,57	12,31	7,46	8,07	7,82
47	6,56	7,23	6,40	6,11	5,43	9,95	8,33	6,98	6,99	7,51	8,04	11,10	8,84	7,72	8,35	9,64	9,92	9,20	10,89	8,17
25	6,28	6,50	6,17	5,69	8,03	11,68	7,84	6,13	6,37	7,48	9,56	12,76	8,62	7,25	7,16	9,33	9,33	9,16	10,46	8,20
510	5,62	5,42	5,20	5,32	8,61	13,82	7,89	6,62	6,09	6,66	10,70	15,35	8,32	8,04	6,41	9,85	10,53	8,62	9,49	8,34
Joint Mean	4,47	5,11	4,33	4,43	6,68	10,72	6,67	5,82	5,93	5,17	7,46	10,74	6,79	6,71	6,91	9,46	10,26	7,53	8,45	

Quality	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Mean
2569	0,90	0,77	0,95	0,83	0,60	0,26	0,95	0,92	0,90	0,87	0,39	0,12	0,93	0,93	0,92	0,61	0,75	0,80	0,75	0,74
127	0,85	0,80	0,82	0,80	0,34	0,13	0,84	0,85	0,93	0,71	0,29	0,17	0,70	0,68	0,72	0,73	0,79	0,57	0,54	0,65
6	0,79	0,46	0,93	0,74	0,52	0,16	0,93	0,88	0,93	0,68	0,24	0,21	0,91	0,85	0,91	0,46	0,58	0,52	0,54	0,64
510	0,72	0,67	0,74	0,68	0,54	0,23	0,74	0,71	0,68	0,65	0,33	0,30	0,73	0,71	0,72	0,69	0,68	0,70	0,70	0,63
25	0,71	0,60	0,74	0,68	0,54	0,33	0,74	0,73	0,73	0,68	0,57	0,34	0,72	0,72	0,72	0,65	0,55	0,51	0,55	0,62
259	0,75	0,69	0,78	0,72	0,42	0,14	0,77	0,70	0,67	0,74	0,46	0,21	0,77	0,71	0,75	0,47	0,39	0,57	0,49	0,59
123	0,63	0,56	0,74	0,59	0,49	0,15	0,75	0,72	0,85	0,63	0,43	0,09	0,73	0,70	0,73	0,49	0,50	0,68	0,68	0,59
9	0,69	0,51	0,74	0,54	0,49	0,13	0,72	0,41	0,59	0,70	0,63	0,28	0,74	0,71	0,72	0,56	0,61	0,61	0,56	0,58
4	0,71	0,64	0,68	0,63	0,20	0,07	0,66	0,66	0,74	0,69	0,29	0,23	0,65	0,67	0,72	0,62	0,63	0,68	0,68	0,57
1278	0,74	0,64	0,78	0,70	0,42	0,19	0,79	0,76	0,76	0,69	0,29	0,14	0,67	0,65	0,66	0,42	0,34	0,59	0,59	0,57
12	0,69	0,64	0,83	0,73	0,49	0,21	0,88	0,81	0,86	0,50	0,32	0,25	0,68	0,70	0,70	0,45	0,38	0,35	0,35	0,57
7	0,72	0,69	0,52	0,69	0,22	0,08	0,62	0,67	0,76	0,69	0,40	0,17	0,45	0,63	0,69	0,67	0,68	0,64	0,61	0,56
47	0,53	0,41	0,71	0,50	0,42	0,23	0,71	0,68	0,66	0,51	0,50	0,31	0,68	0,70	0,71	0,37	0,36	0,53	0,62	0,53
10	0,59	0,48	0,71	0,57	0,35	0,19	0,73	0,31	0,36	0,57	0,34	0,30	0,70	0,72	0,84	0,48	0,57	0,46	0,50	0,51
24	0,64	0,63	0,40	0,62	0,55	0,15	0,44	0,50	0,67	0,60	0,51	0,18	0,37	0,57	0,69	0,45	0,37	0,63	0,63	0,51
5	0,60	0,45	0,71	0,65	0,27	0,15	0,78	0,82	0,67	0,35	0,21	0,22	0,54	0,66	0,66	0,34	0,32	0,30	0,27	0,47
2	0,40	0,30	0,41	0,45	0,26	0,17	0,43	0,56	0,72	0,34	0,32	0,40	0,35	0,49	0,71	0,23	0,14	0,25	0,29	0,38
1	0,42	0,33	0,68	0,40	0,25	0,14	0,64	0,25	0,29	0,31	0,16	0,10	0,66	0,67	0,72	0,25	0,15	0,23	0,20	0,36
8	0,39	0,26	0,31	0,40	0,30	0,15	0,30	0,25	0,34	0,30	0,23	0,06	0,32	0,66	0,65	0,33	0,35	0,26	0,29	0,32
3	0,06	0,05	0,23	0,07	0,15	0,06	0,25	0,29	0,29	0,06	0,17	0,25	0,22	0,55	0,70	0,04	0,03	0,04	0,04	0,19
Joint Mean	0,63	0,53	0,67	0,60	0,39	0,17	0,68	0,62	0,67	0,56	0,35	0,22	0,63	0,68	0,73	0,47	0,46	0,50	0,49	

Table 4.1: Score table of all the models for the different joints.

From the more complete information present in Table 4.1, we can deduce the following pieces of information:

- It seems evident that the more the articulation moves, the harder it is for our model to estimate its position accurately. Indeed we see two columns with obviously poorer results for the three criteria. These are both wrist columns. On the contrary, we see that mostly static articulations, such as the ankles or the body center, have much better results, even in the worst models.
- Models with more cameras do not always outperform models with fewer cameras. Still, there is a tendency for models with more cameras to perform better, with all models based trained with 3 or 4 cameras reaching the top 50% of our ranking. However, it is important to note that models 6 and 9 performed really well. Note that simply adding cameras is not the answer, as shown by the model 47, which is largely outperformed by models 4 and 7 individually.
- The mean criterion is only moderately correlated[30] to the SD (0.58) and the quality (0.68), while these two are not correlated as shown in Figure 4.5. We thus decided to keep the three criteria.
- There are some interesting exceptions. The model 3, while having a really bad mean estimation of the position articulations, is consistent in its estimation with an offset. The model 510 is also quite wrong in his mean estimation of the position of the articulation, with also a large standard deviation but a great quality result. This might mean that some really bad results penalize the global result.
- This experiment does not allow us to give interpretation regarding the placement of the cameras around the subject. Indeed, while the model 2469, trained with spaced cameras, largely outperforms the model 1278, trained with packed cameras, the same can not be said when looking at the models trained with two or three cameras.

4.2.2 Number of cameras during testing

In the previous subsection, we studied the model's global quality, but there may be a difference when using a fixed number of cameras in the combination. We have computed the same tables but with only one-, two-, three- or four-cameras combinations as input when computing the different criteria to see if the number of cameras used when using the model has to be the same as the one during the training.

Model	12	123	25	127	24	510	2569	1278	47	4
Global Score	8	10	16	17	18	23	25	26	29	30

5	259	2	6	9	10	3	7	1	8
37	38	38	39	40	41	41	44	53	57

Table 4.3: Ranking of the 20 models tested with combinations of 1 camera

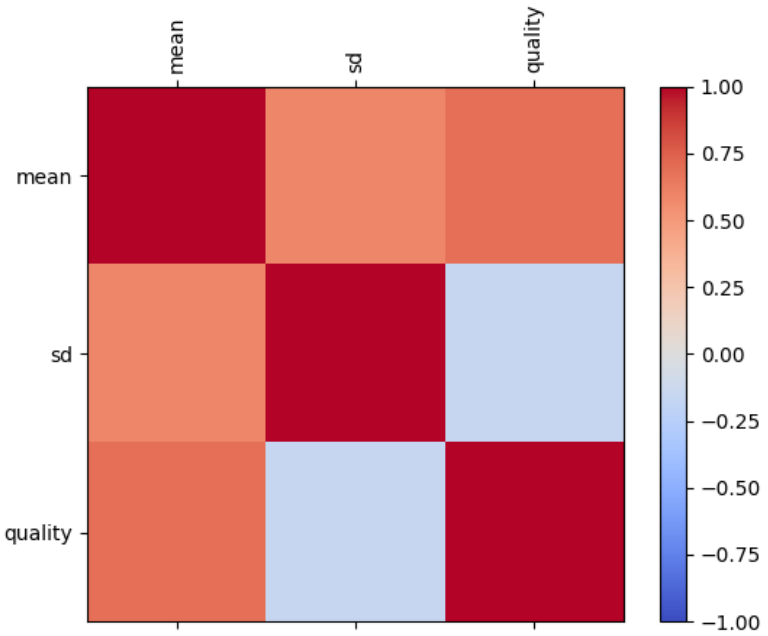


Figure 4.5: Correlation between our three criteria

Model	2569	127	12	123	1278	6	259	25	4	9
Global Score	3	8	17	20	20	23	24	26	29	30

510	7	2	24	10	3	5	47	8	1
32	34	38	39	42	43	45	49	53	55

Table 4.4: Ranking of the 20 models tested with combinations of 2 cameras

Model	2569	127	6	12	9	259	1278	123	4	7
Global Score	3	6	12	21	21	22	23	29	30	32

25	510	2	10	3	24	5	47	8	1
33	36	36	37	43	45	48	49	50	54

Table 4.5: Ranking of the 20 models tested with combinations of 3 cameras

Model	2569	127	6	12	9	1278	259	7	4	2
Global Score	3	8	8	20	20	22	24	30	32	33

123	25	10	510	3	24	8	47	5	1
35	37	37	40	42	43	43	49	49	55

Table 4.6: Ranking of the 20 models tested with combinations of 4 cameras

When looking only at the global ranking, the results are consistent with multiple cameras and basically the same as the fused result we studied earlier. When using only one camera, the results are more random. However, when looking at the detailed results in Table 4.7, we see that there

is an influence on the number of cameras. Indeed, the results of the best models are better with more cameras, while the worst models are even worse with more cameras.

	Mean	SD	Quality
123	10,66	6,26	0,65
12	10,81	5,97	0,57
25	11,03	6,66	0,66
510	11,33	6,81	0,66
24	11,34	6,47	0,55
127	11,79	6,19	0,54
2569	11,91	6,21	0,50
1278	12,02	6,36	0,52
5	12,48	6,80	0,51
47	12,55	6,75	0,54
4	12,67	6,62	0,53
9	12,71	6,76	0,49
259	12,74	6,78	0,52
6	12,96	6,52	0,44
7	12,97	7,32	0,52
10	13,02	6,53	0,46
2	13,51	6,13	0,38
1	13,87	6,84	0,40
8	14,17	6,97	0,34
3	16,61	5,96	0,20

One camera

	Mean	SD	Quality
2569	9,09	5,36	0,71
127	10,03	5,73	0,65
12	10,50	5,98	0,60
6	10,79	6,19	0,59
1278	10,82	6,14	0,61
25	10,96	7,03	0,69
123	10,97	6,54	0,64
9	11,09	6,64	0,60
259	11,23	6,54	0,62
510	11,46	7,19	0,69
7	11,49	6,94	0,61
4	11,68	6,58	0,62
24	11,89	6,66	0,56
10	11,94	6,66	0,54
5	12,74	6,89	0,51
2	12,77	6,04	0,40
47	12,77	7,08	0,58
8	13,76	6,99	0,35
1	14,54	7,07	0,38
3	15,83	5,79	0,20

Two cameras

	Mean	SD	Quality
2569	8,56	5,20	0,75
127	9,99	5,76	0,66
6	10,25	6,10	0,64
12	10,98	6,32	0,58
9	11,28	6,92	0,59
1278	11,32	6,49	0,59
259	11,74	6,93	0,61
7	11,99	7,12	0,58
123	12,10	7,14	0,61
25	12,26	7,85	0,64
4	12,41	6,99	0,59
10	12,50	7,09	0,53
510	12,92	8,00	0,65
2	13,01	6,23	0,39
24	13,03	7,18	0,52
5	13,73	7,38	0,49
8	14,16	7,17	0,33
47	14,22	7,84	0,55
1	15,51	7,60	0,37
3	15,97	5,87	0,19

Three cameras

	Mean	SD	Quality
2569	8,35	5,16	0,76
6	9,97	6,07	0,67
127	10,15	5,88	0,64
12	11,54	6,68	0,55
9	11,79	7,28	0,57
1278	11,93	6,90	0,56
259	12,51	7,39	0,57
7	12,64	7,38	0,54
4	13,31	7,45	0,55
10	13,33	7,56	0,50
2	13,36	6,48	0,37
123	13,36	7,79	0,56
25	13,75	8,72	0,59
24	14,16	7,72	0,48
8	14,55	7,35	0,31
510	14,61	8,86	0,60
5	14,82	7,96	0,45
47	15,84	8,65	0,51
3	16,30	5,99	0,18
1	16,58	8,15	0,35

Four cameras

Table 4.7: Summary of the scores when testing the models with the N-camera combination.

4.2.3 Position of cameras during testing

To study the influence of the position of the cameras when testing the models, we tested the two best-performing models of the previous experiments, models 2569 and 127, with packed or spaced combinations of cameras. The combinations used for the packed cameras are 2-7, 3-8, 4-9, 4-10, 1-6, 1-2-7, 4-5-10, 3-8-9, 1-2-7-8 and 4-5-9-10. For the spaced cameras, the combinations are 1-9, 2-10, 5-8, 3-6, 4-7, 1-5-8, 2-5-9, 4-6-8, 2-5-6-9 and 3-4-6-7.

	Packed				Spaced		
Model	Mean	SD	Quality		Model	Mean	SD	Quality
2569	9,40	5,48	0,69		2569	8,50	5,21	0,75
127	9,74	5,64	0,67		127	10,28	5,87	0,64

Table 4.8: Summary of the scores of models tested with packed or spaced combinations of cameras[1]

We see in Table 4.8 that a model trained with packed cameras performs better when used with packed cameras, and the opposite is true for a model trained with spaced cameras.

[1]The scores of this table can not be compared to the previous results, the testing samples being much smaller here.

4.2.4 Desynchronization of the cameras

Synchronizing multiple cameras can be a challenging hardware difficulty. We will here study the robustness of our model to small desynchronization. We study the effect of desynchronization in three different situations: a desynchronization during the training, a desynchronization during the usage of the model and desynchronization during both steps.

We simulate the desynchronization by adding a random offset to the index of the frame used when fusing the input of different cameras. Sameer Ansari showed in his work[2] that with a naive physical synchronization, the mean desynchronization between two cameras is three frames and the desynchronization can reach up to six frames, while with an efficient synchronization method, the desynchronization is negligible.

We chose an intermediate theoretical synchronization method with a random offset comprised between -3 and +3 frames. There is a 50% chance of having no offset, a 30% chance of having an offset of 1 frame, a 16% chance of having a 2 frames offset and 4 residual percent for the 3 frames offset. This gives a mean offset between two cameras of 1.19 frames with 31% chance of no offset, 35% chance of a 1-frame offset, 21% chance of a 2-frame offset, 9% chance of a 3-frame offset, 3% chance of a 4-frame offset and 1% chance of a 5-frame offset.

It is essential to consider this method's border effects. We might try to get a frame that does not exist, either because it was never recorded (a frame before or after the sequence we use) or because this was discarded by our preprocessing method due to its poor quality. In this case, we simply take the normal frame with no offset.

	Mean	SD	Quality		Mean	SD	Quality
Model	**Both**			Model	**Test**		
2569	**7,65**	**5,04**	**0,78**	2569	8,58	5,21	0,74
123	9,07	5,89	0,72	123	12,62	7,41	0,59
127	13,42	6,72	0,50	127	10,11	5,84	0,65
25	10,27	6,62	0,67	25	12,90	8,20	0,62
1278	11,38	6,60	0,57	1278	11,61	6,68	0,57
Model	**Train**			Model	**None**		
2569	7,66	5,05	0,78	2569	8,59	5,23	0,74
123	9,07	5,89	0,72	123	12,62	7,40	0,59
127	13,45	6,72	0,50	127	10,13	5,83	0,65
25	10,26	6,60	0,67	25	12,89	8,20	0,62
1278	11,37	6,59	0,57	1278	11,61	6,67	0,57

Table 4.9: Summary of the scores of models trained with or without desynchronization and tested with or without desynchronization

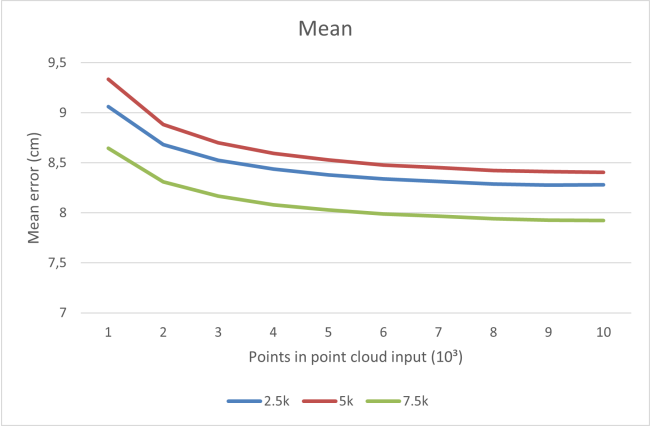
The Table 4.9 is divided into four parts: Both, Train, Test and None. The "None" section is a reminder of the subsection 4.2.1 with the results with no re-synchronization calculated with the same testing sample. The "Both" section presents models trained and tested with a desynchronization and the "Train" and "Test" sections present models respectively trained and tested with a desynchronization.

For models 2569, 123 and 25, we see that the results are better when the model is trained with noise and that the desynchronization does not change the results during the tests. Model 1278 has

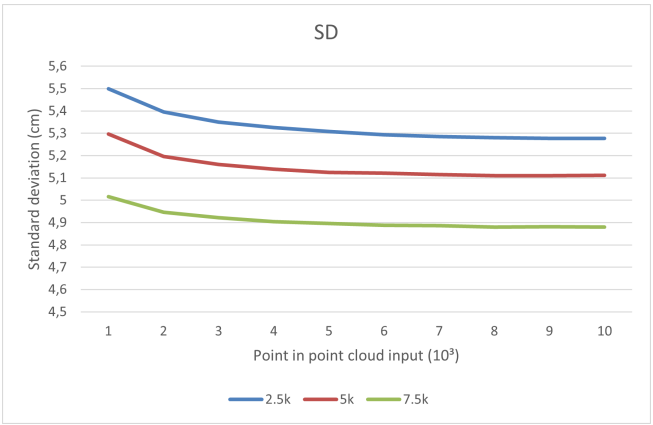
a slight but not significant improvement in the same way. Model 25's results are worse with the desynchronization, but the non-influence of the desynchronization during the test remains valid.

4.2.5 Influence of the resolution of the cameras

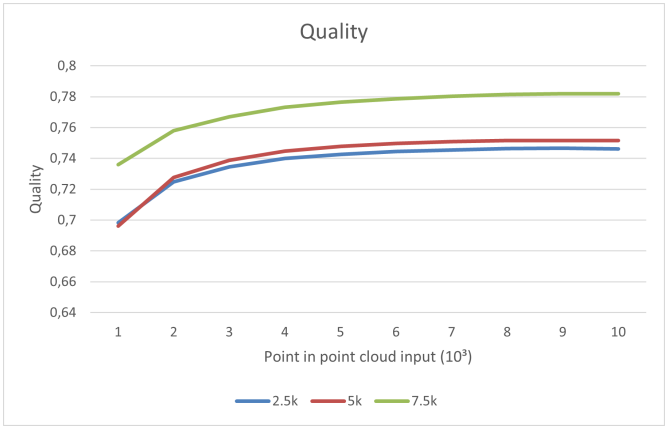
We fixed for all the precedent tests the number of points in the point clouds to 5000 both during the training and the testing. We will now study the influence of this number of points on the quality of the model. We compare three models trained with 2500 points, 5000 points and 7500 points, respectively. These models are then tested with varying input sizes, from 1000 to 10000 points, with 1000 points steps. The test sequence is 40 times shorter in these tests to reduce the computation time. The evolution of our three parameters is shown in Figure 4.6.



(a) Evolution of the mean



(b) Evolution of the standard deviation



(c) Evolution of the quality

Figure 4.6: Evolution of the mean, standard deviation and quality according to the input size.

The model trained with input point clouds of 7500 points performs better in every situation. The model trained with input point clouds of 5000 points performs slightly worse than the one trained with point clouds of 2500 points when looking at the mean criterion. However, the situation is reversed when looking at the standard deviation criterion. Regarding quality, the 5000 points model performs slightly better than the 2500 points one, but the difference is minimal.

Chapter 5

Discussion

Chapter summary: This chapter analyses the results from the previous chapter and discusses the different limitations of our experiments

5.1 Experiments analysis

We did experiments to study the influence of different parameters that may influence the conception of a system used to estimate the position of human joints using ToF cameras.

Regarding the number of cameras, the best results were obtained with a higher number of cameras and we see that, in general, more cameras in the training leads to better results. However, no absolute conclusion can be drawn as the third-best-performing model was trained with only one camera, as seen in subsection 4.2.1. The number of cameras used for the training of the models can be different from the number of cameras used for the test of the model. Still, we do not get evidence of a direct influence of this number on the result in subsection 4.2.2.

Regarding the position of the cameras during the training, even though the best model is a model trained with spread-out cameras, there is no evident best placement, the second best model being one trained with close to each other cameras as seen in subsection 4.2.1. However, we can see in subsection 4.2.3 that it is essential to keep a similar setup during the training and the use of the models. Indeed, a model trained with spaced-out cameras will perform poorly when used with close to each other cameras, and reciprocally.

Regarding synchronizing the different cameras, we see in subsection 4.2.4 that adding a small desynchronization during the training helps the model generalize better. A small desynchronization when using the model is not a problem and does not influence the results. This small desynchronization can be assimilated to a small individual random displacement of the points and can thus also be implemented in a well-synchronized system.

Regarding the resolution of the cameras, a model trained and used with a denser point cloud gives better results. However, there is a limit after 6000 points where the results do not improve or very slowly when using the model. The cameras' resolution can be tuned to get a global point cloud of the desired size after the pre-processing steps depending on the number of cameras used.

Models 3 and 6 behaved with no visible symmetry while being placed in equivalent positions around the subjects. This may be linked to the reduced number of valid frames kept in cameras 3 and 6 by our pre-processing methodology.

Our best-performing model gives results on par with other point-cloud-based approaches on most joints but is really bad on the wrists, elbows and face joints[58].

5.2 Limitations

Our project suffers from some limitations:

- Due to time and computation power limitations, we were not able to replicate our results with multiple random seeds to ensure the stability of our results. At least two additional repetitions of our experiments would be needed to assess the stability of our results, which would take at our current computation speed more than 44 days of continuous running, supposing no time is required to start the codes and neglecting the analysis of the results and the inevitable errors.

- Regression tasks require large and diverse data sets for a good generalization during the training. Our dataset, while large, is unfortunately not diverse enough in the hard poses of the subjects, leading to a lack of generalization in less common movements.

- The training was pushed for too many epochs. We arbitrarily trained our models for 50 epochs and kept this number to avoid losing precious time on retraining models, but using more than 25 epochs was almost useless. Fortunately, the over-fitting was avoided due to the constant data augmentation.

- The PointNet structure is pretty simple and has difficulties generalizing local features in our problem. A more recent but more complex structure might be more adapted, such as PointNet++ [43], which was developed by the same team as PointNet to overcome this issue. This upgraded version applies the PointNet structure recursively on segmented sections of the input point cloud to capture local features of the point cloud.

Chapter 6

Conclusions

Chapter summary: This chapter summarizes our work and dives into the possibilities available to go further in the future

The goal of this thesis was to assess the possibility to use a point cloud generated from the fusion of multiple depth cameras as an input for a neural network to estimate the position of human joints and then to study the influence of different parameters on the quality of the result.

This thesis proves that using the point cloud obtained from the fusion of multiple depth cameras is pertinent when trying to estimate the position of human joints. While the results at the moment are not perfect, they are really promising and, as we show in the next section, have a lot of room for progress.

No absolute conclusion can be drawn regarding the placement and the number of cameras. Nevertheless, regarding the number of cameras, there is a tendency for models trained with more cameras to perform better than models trained with less. Regarding the placement of the cameras, the best results are obtained on a model trained on evenly spaced cameras, but this is not an absolute rule. Despite this lack of conclusive information, we found that it is important to keep a similar setup when using a model as when training it.

A perfect synchronization between the different cameras is not mandatory, and a small desynchronization during the training process can even lead to better results.

The resolution of the cameras does not need to be too high when using the models, a point cloud input of 6000 or 7000 is sufficient, and a higher point cloud size will only bring small improvement. Such a point cloud can be obtained with two cameras with our preprocessing method, with more cameras, the resolution of the cameras can be reduced, allowing easier bandwidth management. However, the higher the resolution during the training, the better.

This thesis allows us to get a beginning of understanding of the requirements of a system estimating the position of human joints using depth cameras and machine learning, but additional work has to be done to alleviate residual uncertainty and get information regarding other parameters.

6.1 To go further

6.1.1 Study other parameters

To define the different parameters that have to be taken into account, we studied the influence of the placement of cameras around the subject, the number of cameras, the good synchronization of the multiple cameras and the density of the point cloud, but there are a lot of other parameters that can be studied, such as:

- Studying the influence of training a network with the fused input of multiple cameras against training a network with different inputs of multiple cameras one at a time. And compare these to training a network for each camera and then fuse the results of the different networks.
- Studying the influence of the color. It is easy to add color to the input of our network and it could be interesting to study its impact on the quality of the result. Indeed, being certain that we can eliminate the color information would greatly facilitate the acquisition of multiple cameras as the data flow would be significantly reduced. With a reduced data flow, the number of ToF cameras that can be synchronized and used at a high frame rate can be considerably increased.
- Evaluating the influence of placement of the cameras on a specific set of movements or in specific plans can be of great use in developing models specific to studying a fixed task. Indeed, the results we studied here were focused on the mean results.
- Go further in the parameters and see if there is further interest in using even more cameras.
- Take into account the cost of the system, both for the hardware cost (cameras and powerful computer) and for the computation time of the models.
- Study the influence of wrongly calibrated cameras both on the training and when using the model

6.1.2 Other neural network structures

We chose to work with an adaptation of the PointNet structure for this thesis as it is one of the first neural networks capable of working directly with 3D point cloud data as input. However, as one of the first neural networks of this kind, it is also one of the simplest and its capacity to extract features from the point cloud is limited. Moreover, this neural network structure was initially developed to perform segmentation and classification tasks, not regression.

There are many other neural network structures that came after PointNet, namely its direct successor, PointNet++ or Minkowski Convolutional Neural Networks. None of them is directly thought for our case, so the best solution might be to start from those and tweak them until the results are the best.

6.1.3 Multiple neural network

Working with multiple neural networks in parallel could be interesting. Each limb could have its own specialized neural network, and a first network would be in charge of the segmentation of the point cloud, a task not yet mastered but in constant amelioration during the last decades[23].

6.1.4 Particle model

Another approach would consist in fitting to a model of the human body. The first problem to come to mind is the sheer variety of shapes that the human body can take, everyone is different. Software exists to generate a patient-specific articulated model that deforms in a realistic manner, such as the STAR model (Sparse Trained Articulated Human Body Regressor)[36]

The second problem that comes to mind is the computation difficulty of fitting a large point cloud to an articulated body. There is an infinity of possible poses for the model, and we would need to test them all until we find the best. Obviously, that is not true for two main reasons. Firstly we will need to develop an algorithm that will converge to the correct solution in a smart way. Secondly, we can use a simple neural network to determine an approximately good starting position of the model.

6.1.5 Further training

An evident method to get better results is to train better our network. To achieve this goal, a large part of the work will be in the dataset and the pre-processing. We will need to use more data and augment the number of frames for the special poses, such as the squats, the high-kick, etc. Additionally, we will need to add to the training data even more variability with active pose, partial occlusion of the body, etc.

Improving data augmentation can also prove to be helpful. Our models would be more resilient and the training would over-fit less rapidly if the point clouds were rotated and translated as a whole. Very small translations of individual points can also be used to augment the data.

Another improvement can be brought by modifying our hyper-parameters. We used the MSE loss function, which give us a good mean convergence, but we could tweak this loss function to suit our problem better. We saw that extremities of the body that moves a lot are slow to converge and keep significant errors after a lot of training. We could modify the loss function to give greater importance to the error in the extremities joints. Other hyper-parameters can also be tweaked by trial and error to get the best results.

6.1.6 Real-time results

Our approach is offline and uses no temporal clues in its joints estimation. A possible improvement would be to work in real-time. The current models process a frame in 15ms. A real-time is possible at 30Hz if an efficient pre-processing pipeline and a parallelization of the processes are developed. This real-time approach would allow the practitioners to give results and analysis instantaneously to the patient.

6.1.7 Generalization to other setups

It is essential to study the robustness of our system to different setups, with different ToF cameras, different illumination, a different disposition of the cameras, etc. This can be done by using other datasets with similar but different setups or with the material available in our laboratory.

Bibliography

[1] Dimitrios S Alexiadis et al. "An Integrated Platform for Live 3D Human Reconstruction and Motion Capturing". In: *IEEE Transactions on Circuits and Systems for Video Technology* 27.4 (2017), pp. 798–813.

[2] Sameer Ansari et al. "Wireless Software Synchronization of Multiple Distributed Cameras". In: *CoRR* abs/1812.09366 (2018).

[3] Richard Baker. "The history of gait analysis before the advent of modern computers". In: *Gait & Posture* 26.3 (2007), pp. 331–342. ISSN: 0966-6362. DOI: https://doi.org/10.1016/j.gaitpost.2006.10.014. URL: https://www.sciencedirect.com/science/article/pii/S0966636206003225.

[4] Jerome Berclaz et al. "Multiple Object Tracking Using K-Shortest Paths Optimization". In: *IEEE transactions on pattern analysis and machine intelligence* 33 (Sept. 2011). DOI: 10.1109/TPAMI.2011.21.

[5] Federica Bogo et al. "Keep It SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe et al. Cham: Springer International Publishing, 2016, pp. 561–578. ISBN: 978-3-319-46454-1.

[6] A Cappozzo et al. "Position and orientation in space of bones during movement: experimental artefacts". In: *Clinical Biomechanics* 11.2 (1996), pp. 90–100. ISSN: 0268-0033. DOI: https://doi.org/10.1016/0268-0033(95)00046-1. URL: https://www.sciencedirect.com/science/article/pii/0268003395000461.

[7] Antonio I Cuesta-Vargas, Alejandro Galán-Mercant, and Jonathan M Williams. "The use of inertial sensors system for human motion analysis". In: *Physical Therapy Reviews* 15.6 (2010). PMID: 23565045, pp. 462–473. DOI: 10.1179/1743288X11Y.0000000006. eprint: https://doi.org/10.1179/1743288X11Y.0000000006. URL: https://doi.org/10.1179/1743288X11Y.0000000006.

[8] R. Ducroquet, J. Ducroquet, and P. Ducroquet. *Walking and Limping: A Study of Normal and Pathological Walking*. Lippincott, 1968. URL: https://books.google.be/books?id=PahsAAAAMAAJ.

[9] Sergi Foix, Guillem Alenyà, and Carme Torras. "Lock-in Time-of-Flight (ToF) cameras: A survey". In: *Sensors Journal, IEEE* 11 (Oct. 2011), pp. 1917–1926. DOI: 10.1109/JSEN.2010.2101060.

[10] Ikhsanul Habibie et al. "In the Wild Human Pose Estimation Using Explicit 2D Features and Intermediate 3D Representations". In: *CoRR* abs/1904.03289 (2019).

[11] D. Heitzmann et al. "Markerless versus marker-based motion analysis in subjects with lower limb amputation: A case series". In: *Gait & Posture* 97 (2022). ESMAC 2022 Abstracts, S95–S96. ISSN: 0966-6362. DOI: https://doi.org/10.1016/j.gaitpost.2022.07.067. URL: https://www.sciencedirect.com/science/article/pii/S096663622200265X.

[12] Fabian Horst et al. “Explaining the unique nature of individual gait patterns with deep learning”. In: *Scientific Reports* 9.1 (Feb. 2019), p. 2391. ISSN: 2045-2322. DOI: 10.1038/s41598-019-38748-8. URL: https://doi.org/10.1038/s41598-019-38748-8.

[13] Catalin Ionescu et al. “Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.7 (July 2014), pp. 1325–1339.

[14] Karim Iskakov et al. “Learnable Triangulation of Human Pose”. In: *CoRR* abs/1905.05754 (2019).

[15] Abhinav Jain et al. “Overview and Importance of Data Quality for Machine Learning Tasks”. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD ’20. Virtual Event, CA, USA: Association for Computing Machinery, 2020, pp. 3561–3562. ISBN: 9781450379984. DOI: 10.1145/3394486.3406477. URL: https://doi.org/10.1145/3394486.3406477.

[16] Hanbyul Joo et al. “Panoptic Studio: A Massively Multiview System for Social Interaction Capture”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).

[17] Robert M. Kanko et al. “Assessment of spatiotemporal gait parameters using a deep learning algorithm-based markerless motion capture system”. In: *Journal of Biomechanics* 122 (2021), p. 110414. ISSN: 0021-9290. DOI: https://doi.org/10.1016/j.jbiomech.2021.110414. URL: https://www.sciencedirect.com/science/article/pii/S0021929021001949.

[18] Robert M. Kanko et al. “Concurrent assessment of gait kinematics using marker-based and markerless motion capture”. In: *Journal of Biomechanics* 127 (2021), p. 110665. ISSN: 0021-9290. DOI: https://doi.org/10.1016/j.jbiomech.2021.110665. URL: https://www.sciencedirect.com/science/article/pii/S0021929021004346.

[19] Robert M. Kanko et al. “Inter-session repeatability of markerless motion capture gait kinematics”. In: *Journal of Biomechanics* 121 (2021), p. 110422. ISSN: 0021-9290. DOI: https://doi.org/10.1016/j.jbiomech.2021.110422. URL: https://www.sciencedirect.com/science/article/pii/S0021929021002025.

[20] Nitish Shirish Keskar et al. “On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima”. In: *CoRR* abs/1609.04836 (2016). arXiv: 1609.04836. URL: http://arxiv.org/abs/1609.04836.

[21] Diederik Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *International Conference on Learning Representations* (Dec. 2014).

[22] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. “Self-Supervised Learning of 3D Human Pose using Multi-view Geometry”. In: *CoRR* abs/1903.02330 (2019).

[23] Damian Krawczyk and Robert Sitnik. “Segmentation of 3D Point Cloud Data Representing Full Human Body Geometry: A Review”. In: *Pattern Recognition* 139 (2023), p. 109444. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2023.109444. URL: https://www.sciencedirect.com/science/article/pii/S0031320323001449.

[24] Lianhua Li et al. “Multi-camera interference cancellation of time-of-flight (TOF) cameras”. In: Sept. 2015. DOI: 10.1109/ICIP.2015.7350860.

[25] Sijin Li and Antoni Chan. “3D Human Pose Estimation from Monocular Images with Deep Convolutional Neural Network”. In: vol. 9004. Nov. 2014, pp. 332–347. ISBN: 978-3-319-16807-4. DOI: 10.1007/978-3-319-16808-1_23.

[26] Matthew Loper et al. "SMPL: A skinned multi-person linear model". In: *ACM Transactions on Graphics* 34.6 (2015). DOI: 10.1145/2816795.2818013. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84995874936&doi=10.1145%2f2816795.2818013&partnerID=40&md5=fbeac21b0708b07e1d1538ce1038ba31.

[27] Diogo C. Luvizon, David Picard, and Hedi Tabia. "2D/3D Pose Estimation and Action Recognition using Multitask Deep Learning". In: *CoRR* abs/1802.09232 (2018).

[28] Christian Maiwald et al. "The effect of intracortical bone pin application on kinetics and tibiocalcaneal kinematics of walking gait". In: *Gait & Posture* 52 (2017), pp. 129–134. ISSN: 0966-6362. DOI: https://doi.org/10.1016/j.gaitpost.2016.10.023. URL: https://www.sciencedirect.com/science/article/pii/S0966636216306117.

[29] Dalton D. Moore et al. "Validating markerless pose estimation with 3D X-ray radiography". In: *Journal of Experimental Biology* 225.9 (May 2022), jeb243998. ISSN: 0022-0949. DOI: 10.1242/jeb.243998. eprint: https://journals.biologists.com/jeb/article-pdf/225/9/jeb243998/2144107/jeb243998.pdf. URL: https://doi.org/10.1242/jeb.243998.

[30] M M Mukaka. "Statistics corner: A guide to appropriate use of correlation coefficient in medical research". en. In: *Malawi Med J* 24.3 (Sept. 2012), pp. 69–71.

[31] Christopher Nester et al. "Foot kinematics during walking measured using bone surface markers". In: *Journal of biomechanics* 40 (Feb. 2007), pp. 3412–23. DOI: 10.1016/j.jbiomech.2007.05.019.

[32] Alejandro Newell, Kaiyu Yang, and Jia Deng. "Stacked Hourglass Networks for Human Pose Estimation". In: *CoRR* abs/1603.06937 (2016).

[33] Bruce Xiaohan Nie, Ping Wei, and Song-Chun Zhu. "Monocular 3D Human Pose Estimation by Predicting Depth on Joints". In: vol. 2017-October. 2017, pp. 3467–3475. DOI: 10.1109/ICCV.2017.373. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85035309520&doi=10.1109%2fICCV.2017.373&partnerID=40&md5=efc9a951fe3347ff39ef948e208824e4.

[34] Eline M. Nijmeijer et al. "Concurrent validation of the Xsens IMU system of lower-body kinematics in jump-landing and change-of-direction tasks". In: *Journal of Biomechanics* 154 (2023), p. 111637. ISSN: 0021-9290. DOI: https://doi.org/10.1016/j.jbiomech.2023.111637. URL: https://www.sciencedirect.com/science/article/pii/S0021929023002063.

[35] Matthew A. Nurse and Benno M. Nigg. "The effect of changes in foot sensation on plantar pressure and muscle activity". In: *Clinical Biomechanics* 16.9 (2001), pp. 719–727. ISSN: 0268-0033. DOI: https://doi.org/10.1016/S0268-0033(01)00090-0. URL: https://www.sciencedirect.com/science/article/pii/S0268003301000900.

[36] Ahmed A A Osman, Timo Bolkart, and Michael J. Black. "STAR: A Sparse Trained Articulated Human Body Regressor". In: *European Conference on Computer Vision (ECCV)*. 2020, pp. 598–613. URL: https://star.is.tue.mpg.de.

[37] Manuel Palermo et al. "A multi-camera and multimodal dataset for posture and gait analysis". In: *Scientific Data* 9.1 (Oct. 2022), p. 603. ISSN: 2052-4463. DOI: 10.1038/s41597-022-01722-7. URL: https://doi.org/10.1038/s41597-022-01722-7.

[38] Sungheon Park, Jihye Hwang, and Nojun Kwak. "3D Human Pose Estimation Using Convolutional Neural Networks with 2D Pose Information". In: *Computer Vision – ECCV 2016 Workshops*. Ed. by Gang Hua and Hervé Jégou. Cham: Springer International Publishing, 2016, pp. 156–169. ISBN: 978-3-319-49409-8.

[39] Georgios Pavlakos et al. “Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose”. In: *CoRR* abs/1611.07828 (2016).

[40] Georgios Pavlakos et al. “Harvesting Multiple Views for Marker-less 3D Human Pose Annotations”. In: *CoRR* abs/1704.04793 (2017).

[41] Arthur Leslie Peck, Edward Seymour Forster, et al. *Parts of animals; Movement of animals; Progression of animals/De partibus animalium.* Tech. rep. Harvard University Press; 1937.

[42] Charles Ruizhongtai Qi et al. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation”. In: *CoRR* abs/1612.00593 (2016). arXiv: 1612.00593. URL: http://arxiv.org/abs/1612.00593.

[43] Charles Ruizhongtai Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space”. In: *CoRR* abs/1706.02413 (2017).

[44] Haibo Qiu et al. “Cross View Fusion for 3D Human Pose Estimation”. In: *CoRR* abs/1909.01203 (2019).

[45] Ben Sapp and Ben Taskar. “MODEC: Multimodal Decomposable Models for Human Pose Estimation”. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 3674–3681. DOI: 10.1109/CVPR.2013.471.

[46] Xiao Sun et al. “Integral Human Pose Regression”. In: *CoRR* abs/1711.08229 (2017).

[47] Bugra Tekin et al. “Fusing 2D Uncertainty and 3D Cues for Monocular Body Pose Estimation”. In: *CoRR* abs/1611.05708 (2016).

[48] Denis Tomè, Chris Russell, and Lourdes Agapito. “Lifting from the Deep: Convolutional 3D Pose Estimation from a Single Image”. In: *CoRR* abs/1701.00295 (2017).

[49] Denis Tomè et al. “Rethinking Pose in 3D: Multi-stage Refinement and Recovery for Marker-less Motion Capture”. In: *CoRR* abs/1808.01525 (2018).

[50] Jinbao Wang et al. “Deep 3D human pose estimation: A review”. In: *Computer Vision and Image Understanding* 210 (2021), p. 103225. ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2021.103225. URL: https://www.sciencedirect.com/science/article/pii/S1077314221000692.

[51] Shih-En Wei et al. “Convolutional Pose Machines”. In: *CoRR* abs/1602.00134 (2016). arXiv: 1602.00134. URL: http://arxiv.org/abs/1602.00134.

[52] Felix Wermke and Beate Meffert. “Interference Model of Two Time-Of-Flight Cameras”. In: Oct. 2019, pp. 1–4. DOI: 10.1109/SENSORS43011.2019.8956892.

[53] Laura S. Weyrich et al. “Neanderthal behaviour, diet, and disease inferred from ancient DNA in dental calculus”. In: *Nature* 544.7650 (Apr. 2017), pp. 357–361. ISSN: 1476-4687. DOI: 10.1038/nature21674. URL: https://doi.org/10.1038/nature21674.

[54] D J Wilson et al. “Accuracy of digitization using automated and manual methods”. en. In: *Phys Ther* 79.6 (June 1999), pp. 558–566.

[55] Markus Windolf, Nils Götzen, and Michael Morlock. “Systematic accuracy and precision analysis of video motion capturing systems—exemplified on the Vicon-460 system”. In: *Journal of Biomechanics* 41.12 (2008), pp. 2776–2780. ISSN: 0021-9290. DOI: https://doi.org/10.1016/j.jbiomech.2008.06.024. URL: https://www.sciencedirect.com/science/article/pii/S0021929008003229.

[56] Tishya A.L. Wren, Pavel Isakov, and Susan A. Rethlefsen. "Comparison of kinematics between Theia markerless and conventional marker-based gait analysis in clinical patients". In: *Gait & Posture* 104 (2023), pp. 9–14. ISSN: 0966-6362. DOI: https://doi.org/10.1016/j.gaitpost.2023.05.029. URL: https://www.sciencedirect.com/science/article/pii/S0966636223001443.

[57] Ying Wu, Ting Yu, and Gang Hua. "A Statistical Field Model for Pedestrian Detection". In: vol. 1. July 2005, 1023–1030 vol. 1. ISBN: 0-7695-2372-2. DOI: 10.1109/CVPR.2005.49.

[58] Tianxu Xu et al. "A Review: Point Cloud-Based 3D Human Joints Estimation". In: *Sensors* 21.5 (2021). ISSN: 1424-8220. DOI: 10.3390/s21051684. URL: https://www.mdpi.com/1424-8220/21/5/1684.

[59] Li Zhang, Bo Wu, and Ram Nevatia. "Detection and Tracking of Multiple Humans with Extensive Pose Articulation". In: Nov. 2007, pp. 1–8. ISBN: 978-1-4244-1631-8. DOI: 10.1109/ICCV.2007.4408940.

[60] Xingyi Zhou et al. "Deep Kinematic Pose Regression". In: *Computer Vision – ECCV 2016 Workshops*. Ed. by Gang Hua and Hervé Jégou. Cham: Springer International Publishing, 2016, pp. 186–201. ISBN: 978-3-319-49409-8.

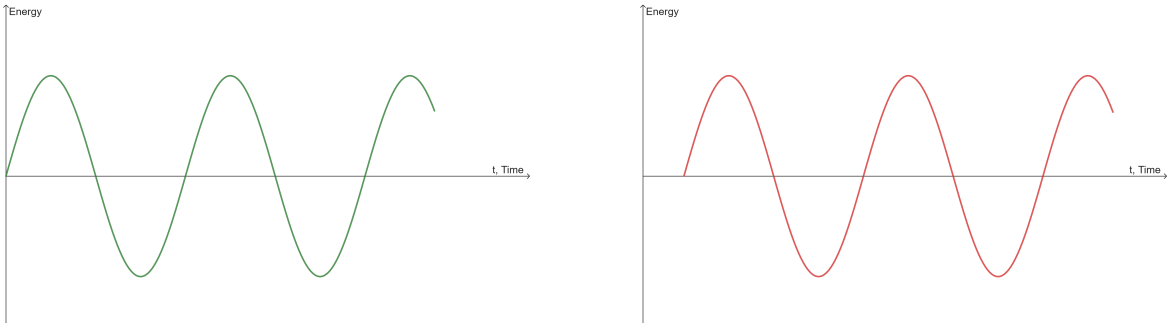
Appendix A

Time of Flight cameras

Time of Flight (ToF) cameras are cameras that give depth information for each pixel instead of color information. Each pixel encodes the distance of the point in the image. The general principle is that the camera emits infrared light, and the sensor measures the phase delay of the light that reflects on the scene.

A.1 Measurement of the depth

The infrared light is emitted at a certain known modulated frequency f. It then bounces on the scene and comes back to the sensor with a certain phase shift. The emitted signal is given by Equation A.1, and the received signal is given by Equation A.2 and they are represented in Figure A.1



(a) Emitted signal (b) Received signal

Figure A.1: Emitted and received signal with phase shift

$$s(t) = Acos(\omega t), \quad \omega = 2\pi f \tag{A.1}$$

$$r(t) = acos(\omega(t-\theta)) + B \tag{A.2}$$

The cross-correlation between the signals gives us Equation A.3

$$\begin{aligned} C(x) &= \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} r(t)s(t+x)dt \\ C(\psi) &= \frac{Aa}{2} cos(\phi+\psi) + B, \quad \phi = 2\pi f\theta \quad \psi = 2\pi fx \end{aligned} \tag{A.3}$$

Where ϕ is the phase, A and a are the amplitudes of the signals, and B is an offset due to the ambient illumination. Every pixel of the sensor will sample the measurement at four moments of equal distances, $\psi_1 = 0$, $\psi_2 = 90$, $\psi_3 = 180$ and $\psi_4 = 270$. From the values of $C(\psi_i)$, we obtain the following :

$$\phi = 2\pi f\theta = arctan(\frac{C(\psi_4) - C(\psi_2)}{C(\psi_1) - C(\psi_3)}) \tag{A.4}$$

$$a = \frac{1}{2A}\sqrt{(C(\psi_3) - C(\psi_4))^2 + (C(\psi_1) - C(\psi_2))^2} \tag{A.5}$$

$$B = \frac{1}{4}(C(\psi_1) + C(\psi_2) + C(\psi_3) + C(\psi_4)) \tag{A.6}$$

The depth d can be obtained with Equation A.7, where c is the speed of light. There is a maximum to the depth that can be measured with no ambiguity, as θ is defined between 0 and 2π. With a modulation frequency of 30MHz, the maximum distance is 5 meters.

$$d = \frac{1}{2}c\ \theta = \frac{c}{2f}\frac{\theta}{2\pi} \tag{A.7}$$

A.2 Errors and interference with ToF cameras

As any cameras, ToF cameras are prone to errors. There are two big types of errors in ToF cameras[9], systematic errors and non-systematic errors. Systematic errors are:

- Depth distortion: the emitted light is never exactly as supposed due to irregularities in the modulation process;
- Integration-time-related: the integration time used to measure the depth can change the result, being more precise with a longer integration time, but more susceptible to noise;
- Built-in pixel related: Every pixel is individually manufactured and can vary slightly from the others. Two neighbor pixels can give different outputs while really measuring the same distance;
- Amplitude-related: the amplitude of the received light can vary for two objects at the same distance, depending on the reflectivity of the objects;
- Temperature-related: the photo-detectors of the camera are sensible to temperature. When working, the electronics heat up and the output can vary.

Those errors can be taken into account during the calibration and limited, at least inside a certain range of parameters for the illumination, the temperature and the distances.

The non-systematic errors are:

- Signal-to-noise ratio distortion: if a scene is not illuminated uniformly, the signal-to-noise ratio will vary across the scene, leading to a less illuminated part being much more sensitive to noise. The quality of the measurement is not uniform;
- Multiple light receptions: light can take multiple paths before arriving at the captor due to edges or concavities in the objects in the scene;

- Light scattering: light can scatter and bounces back to the captor from near objects before the direct light has time to come back from the further object it was supposed to measure;
- Motion blurring: as for normal cameras, fast movement during the capture of the image leads to blur. The longer the integration time, the more sensible the camera is to motion blurring.

Those errors can not be avoided but can be limited through post-treatment of the data, good illumination, good scene preparation and a convenient integration time.

Another problem that can appear when working with multiple ToF cameras around the same working space is interference between those ToF cameras. The first step to limit interference is to use a different modulation frequency for every ToF camera. As shown by Felix Wermke and Beate Meffert in 2020 [52], the interference is higher when the modulation frequencies are higher and the nearest those are to each other.

Some work has been done to cancel those interference, for example by Li Lianhua and his team [24]. This method limits the interference between two ToF cameras of known modulation frequency, but is at the moment still limited to two cameras, further investigations are needed to generalize this solution to more cameras.