3D motion reconstruction with deep learning methods : application to motor disabilities

Mansour Tchenegnon

COLLEGE MATHSTIC
DOCTORAL BRETAGNE
BRETAGNE OCEANE

Université Bretagne Sud

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE BRETAGNE SUD

ÉCOLE DOCTORALE N° 644
Mathématiques et Sciences et Technologies de l'Information et de la Communication en Bretagne Océane
Spécialité : *Informatique et Architectures numériques*

Par
Mansour TCHENEGNON

3D motion reconstruction with deep learning methods - application to motor disabilities

Thèse présentée et soutenue à Université Bretagne Sud (Vannes), le 26 juin 2024
Unité de recherche : IRISA, CNRS, UMR 6074

Rapporteurs :

Alexandre MEYER Maître de Conférences (HDR), Université Lyon 1
Ludovic HOYET Chargé de Recherche (HDR), INRIA Rennes

Composition du Jury :

Président :	Richard KULPA	Professeur des Universités, Université Rennes 2
Examinateur :	Jean-Philippe VANDEBORRE	Professeur des Universités, Institut Mines-Telecom Nord Europe
Dir. de thèse :	Sylvie GIBET	Professeure des Universités, Université Bretagne Sud
Co-enc. de thèse :	Thibaut Le NAOUR	Chercheur CEO, Motion-Up
Co-enc. de thèse :	Willy ALLEGRE	Directeur Technique CoWork'HIT, Centre de Kerpape

Invité(s) :

Pradon DIDIER Chercheur (HDR), Hôpital Raymond-Poincaré – Hôpitaux universitaires Paris Ile-de-France Ouest

Table of Contents

Chapter 1

Introduction

Contents

1.1 Context

Human motion data refers to digital records of movements performed by people in a large variety of contexts. This data can be acquired and represented in several ways, and includes a wide range of relevant movements, from sports and artistic movements, to trauma-specific movements, to movements performed in everyday life. This motion data is therefore a valuable resource that can be used in multiple applications, including:

— In *Entertainment*, motion data can animate humanoid creatures in movies, characters in video games and virtual reality environments. In the latter context, motion data can also be used to control characters interactively through gesture recognition or motion detection.

— In *Robotics*, robots can be remotely controlled through motion tracking or recognition. Furthermore, human-robot interaction uses motion data to develop robots capable of understanding and responding to human gestures.

— There are also applications in the field of *Health care* where motion data is analyzed to design rehabilitation programs, to monitor patients' movements and track their progress. Fall detection systems are also developed, with the possibility of sending alerts to caregivers or emergency services.

— *Sports science* uses biomechanical analysis to study athletes' movements and provide feedback to optimize their performance, prevent injuries, and propose customized training programs.

— *Educational tools* can be developed to learn from interaction in virtual and augmented reality. One example is an immersive application developed to learn first-aid gestures, with real-time feedback [1]. Motion data can also be used to develop immersive surgery applications, particularly for learning surgical techniques. Another example is data-driven sign language learning, using avatar technology [87].

Several artificial intelligence (AI) methods have revolutionized data processing in various domains over the last few years. Machine learning and, more recently, deep learning techniques make it easier and more efficient to process data for tasks such as prediction, recognition, or the generation of new data. This also applies to motion data. As most of these methods rely on large amount of data to operate efficiently, especially deep learning ones, this raises the question of how to obtain this data.

Given the many ways in which motion data can be used, its acquisition has therefore become an important issue for which numerous solutions have been developed. In this thesis, we focus on kinematic data of human movements, notably their spatiotemporal aspects. We focus more specifically on the kinematic data of human movements, in particular their spatiotemporal aspects. In the research fields considered here, this data is traditionally represented geometrically and kinematically, in the form of positions and rotations of skeletal joints.

Before describing our goals, we briefly present below the main advantages and drawbacks of the motion acquisition technologies exploited to record motion data.

1.1.1 Motion Capture Systems

For the purpose of recording the movements of people, various motion capture (*MoCap*) technologies have been developed to track and capture motion:

— Optical *MoCap* systems that are either passive or active. Passive systems use reflexive markers attached to the body and multiple infrared cameras. The principle is to measure the position of the markers while tracking them over time. Active systems on the other hand, use LED markers that emit their own light. A software is used to identify the positions of the different markers.

1. `https://www.motion-up.com/`

— Non-optical *MoCap* systems that use magnetic, inertial or mechanical sensors to track the body and compute the motion. These systems can be intrusive and are generally less accurate than optical systems.

MoCap systems, especially optical ones with passive markers, are the most widely used for their reliability and accuracy. They enable high-precision capture, down to the millimeter, and at high frequencies, in excess of 120 Hz. For some applications in Computer Animation, these systems lead to data-based synthesis methods being favored over other synthesis methods (descriptive or procedural) in order to produce better, more natural animations. For example, complex movements and technical gestures such as acrobatic flip in football video game can be captured in a physically accurate manner and used to animate characters.

There are, however, some limitations in using these systems. Firstly, the equipment is rather costly and its installation is usually constrained by specific requirements such as the environment in which it is to be used. In addition, these systems require human resources for their usage and may sometimes require post-processing operation that can be time-consuming. Thankfully, with the help of AI methods, new systems manage to reduce the time needed for post-processing [2]. There are also specific situations where these systems are difficult to use. For example, in situations where there are many people interacting with each, the installation time increases as well as the difficulty of labeling and tracking the markers without error especially when individuals get closer during interactions. There are also cases of interactions with the environment in activities such as climbing a wall, playing an instrument, or activities of wheelchair users, where external objects may obstruct the markers (occlusion problems), creating gaps in the recordings, thus hindering the tracking and generating errors that are difficult to correct, even with post-processing.

Although, *MoCap* systems are limited in certain situations and sometimes difficult to use due to conditions such as high cost, environment requirement, technical skills needed, etc., they remain a preferred option to record motion data.

1.1.2 Markerless Systems as an Alternative to MoCap Systems

Markerless systems are a type of optical systems to record motion in which the subject does not need to wear special equipment (such as a suit equipped with markers) for

2. Automatic labeling allows to reduce the post-processing time. See more on `https://docs.optitrack.com/motive/labeling`

tracking.

Some of these systems use multiple video cameras installed in a studio for tracking and capturing motion [3]. The cameras are set up in different positions in order to obtain various viewpoints. The sequences of images produced by the cameras are used to computes the motion as a sequence 3D skeletal postures of people.

Other systems are based on RGB-D cameras, such as Kinect that capture both color and depth images. The two images are used to generate *voxels* (3D pixels) that are used to produce 3D skeletal postures over time. These systems can capture real-time motion but are usually noisy because they use a single-view camera.

Driven by researches and techniques in Computer Vision such as human pose estimation or human motion reconstruction, new systems for recording motion from video have recently been developed. These systems use deep learning methods to estimate 3D skeletal postures of a single person or several people in the scene. There are single-view systems that estimate 3D skeletal postures from a single camera or multi-view systems that use multiple cameras with different viewpoints. The best existing methods reconstruct motion with an error margin of 3 to 4 cm for videos recorded at a frequency rate up to 50 Hz. This type of systems is easy to deploy and very suitable for real-time motion tracking.

Advantages and limitations of markerless systems with AI models

Hypothetically speaking, markerless systems, more precisely those that use AI models to reconstruct motion sequence from video should be able to overcome some of the problems encountered by MoCap systems. First of all, they are generally low cost, easy to use and not intrusive as they do not require people to wear any equipment. Secondly, with intensive training, AI models should be able to learn to reconstruct motion in these situations we mentioned as long as a huge amount of data is available. In addition, these systems usually enable real-time acquisition.

Despite the advantages they offer, these systems are not yet sufficiently accurate to be preferred to MoCap systems. Indeed, in terms of capture accuracy, optical MoCap systems are better (<1mm vs. $\geq$ 3cm error). Moreover, compared to optical MoCap systems which can capture at high frequencies (higher than 200Hz), the temporal quality of motion reconstructed from video at a frequency rate of 50Hz is lower. In addition, most existing AI methods focus mainly on the accuracy of the reconstruction and less on the

3. Example of CMU Panoptic Studio `https://www.cs.cmu.edu/~hanbyulj/panoptic-studio/`

temporal quality. Finally, the existing models do not yet fully solve occlusion problems due to external objects.

1.2 Motivations and Objectives

We are interested in capturing and reconstructing motion from video with an emphasis on the temporal quality of the reconstruction. We intend to later apply the reconstruction system to cases of motor disability that require to use a wheelchair and where the use of MoCap systems is limited.

Numerous applications can be developed from motion tracking and capture, especially in order to improve the autonomy and daily life of people with motor disability. It is possible to develop home monitoring systems that can track and analyze people's movements from home. It is useful for activity recognition and behavior analysis. In particular, such systems can detect anomalies, such as long-time inactivity and falls, and then send an alert for help. They can also analyze the daily activities of people in order to adapt the environment (wheelchair, kitchen and bathroom equipment, door, etc.) according to the analysis results, thus facilitating their autonomy. Finally, if the system is sufficiently accurate, it can be used for other motion capture applications.

As mentioned above, the existing *MoCap* technologies are the most effective way to capture motion data with high accuracy. However, these systems are used in specific environments and sometimes have to meet strict requirements to be effectively used, which means that these technologies cannot be used in every situation. This is particularly true for capturing the movements of people with motor disabilities. For example, it can be difficult to obtain the reference pose (so called standing T-pose) necessary for post-processing. Also, there can be occlusion problems resulting from the usage of a mobile wheelchair that occludes some markers from the cameras. Furthermore, these solutions, very costly and time-consuming are not for anytime use. Lastly, it is not ideal to install such system at home and coerce people to wear a suit with markers for a long period of time, which is intrusive. So, these systems cannot be used for real-time applications such as home monitoring.

A strong hypothesis underlying our work is that markerless capture systems based on video cameras coupled with AI models are suitable for motion capture and reconstruction in situations where MoCap systems are limited. However, existing AI models for motion reconstruction are not yet sufficiently accurate, and generally neglect certain aspects of

motion such as temporal continuity. In addition, despite the existence of AI models for reconstructing human movements from videos, no system for capturing movements in situations of motor disability has yet been designed. This can be explained by the fact these AI models are trained on massive amount of specific data, which prevents them from generalizing. In other words, these models perform well on motion data directly related to those with which they have been trained (identical or similar databases), but perform less well on new categories of motion data. Finally, a large amount of data is generally required to train and evaluate models. To date, however, there are very few databases available on the movements of people with motor disabilities using wheelchairs.

This leads us to define our thesis objectives as follows:

> In this thesis, we aim to propose a markerless system for easily capturing motion with improved spatiotemporal quality. We propose to use deep learning methods to reconstruct motion from video, emphasizing its temporal continuity while preserving its skeletal structure. We then intend to apply this system to the specific case of motor disability. The ultimate goal is for our system to easily capture, track and reconstruct motion data, at low cost, for real-time applications such as daily activity analysis and home monitoring assistance in connected apartments for people with disabilities.

From our objective, two research axes are considered:

1. **Motion reconstruction from video.** To offer an easy-to-use, portable system, we have chosen to reconstruct motion from a single-view camera. For the motion reconstruction approach, we opted for a 2-steps methodology. First, we use a model to estimate motion from video as a sequence of 3D poses. Then, we propose an algorithm to correct the shortcomings in the sequence. This reconstructs a motion of better quality by improving the temporal characteristics while preserving the skeletal structure throughout the sequence. With this approach, we intend to particularly focus on the temporal quality of motion reconstruction, an aspect that most existing AI models neglect, especially with the large difference in frequency rate between optical MoCap systems and markerless video-based systems.
2. **Application to motor disabilities.** In order to train AI models capable of reconstructing motion in motor disability situations, we need a sufficient amount of motion data in these situations, that is video data and corresponding highly

precise 3D motion data. To obtain such data, we have chosen to use an original approach consisting in generating 2D videos from a small amount of 3D movements captured with a MoCap system. This data-driven synthesis approach increases the initial database. Indeed, a larger variety of data can be produced, notably by recording different points of view in the virtual environment, or by using virtual characters with different morphologies.

1.3 Contributions

In line with our objectives, the different achievements realized in this thesis are as follows:

1. A study of motion representation and deep learning methods and their usage in motion reconstruction from video data. Through this study, we analyze existing deep learning solutions proposed in the state of the art and propose our own solution. We also propose a spatiotemporal loss function to train AI models in 3D motion reconstruction task. In addition, we propose spatial and temporal metrics to evaluate the quality of reconstructed motion.
2. A motion correction system using deep learning methods and Laplacian motion modeling. After movements have been reconstructed from video using existing deep learning methods, they are corrected by this system to improve the spatial and temporal quality of these movements.
3. A new motion capture database consisting of movements in motor disability situations. This database contains actions from daily activities of people using mobile wheelchair. From the captured motion data, we generate animated videos, then, we carry out some experiments to test previously developed motion reconstruction approach on these videos.

1.4 Thesis Outline

This PhD thesis is divided into 8 chapters organized as follows:

— A *Literature Review* part regrouping 2 chapters. Chapter 2 is related to existing deep learning methods for motion reconstruction from video while Chapter 3 presents existing motion databases.

— A *Motion Processing and Deep Learning* part summarizing our work on the motion reconstruction task in 3 chapters. Chapter 4 presents the human motion representation we used throughout our work. Chapter 5 describes our solution for the first step of our approach for motion reconstruction from videos. Finally, Chapter 6 presents the motion correction system we developed to improve the quality of motion reconstructed from video.

— A *Motion Database and Application to Motor Disability* part that presents, in a single chapter (Chapter 7), a new database of motion captured in disability case situations as well as deep learning experiments realized on these data.

1.5 Scientific Publications

The following publications were produced based on the work presented in this thesis:

1. Mansour Tchenegnon, Thibaut Le Naour, Sylvie Gibet. "CVM-Net: Motion Reconstruction from a Single RGB Camera with a Fully Supervised DCNN". In: *Les Journées Françaises de l'Informatique Graphique*, Nov 2021, Sophia Antipolis, France. (hal-03536041)
2. Mansour Tchenegnon, Sylvie Gibet, Thibaut Le Naour. "A New Spatio-Temporal Loss Function for 3D Motion Reconstruction and Extended Temporal Metrics for Motion Evaluation". In: *European Conference on Computer Vision (ECCV 2022), Workshop on What is Motion for?*, Oct 2022, Tel Aviv, Israel. (hal-03966941)
3. Mansour Tchenegnon, Sylvie Gibet, Thibaut Le Naour. "MoCoSys: Human Motion Correction based on Deep Learning Coupled with 3D+t Laplacian Motion Representation". 2023. (Under Review, Submitted to *The Visual Computer* on November 2023)
4. Mansour Tchenegnon, Thibaut Le Naour, Willy Allègre, Sylvie Gibet. "HandiMotion: Corpus de Données de Mouvements Capturés en Situation de Handicap". June 2024, Paris, France. (Accepted at *Handicap 2024* Conference organised by IFRATH)

Part I

Literature Review

Chapter 2

Motion Reconstruction and Deep Learning

Contents

We are interested in the idea of reconstructing motion from video. Advances in deep learning techniques give hope to the making of simple systems for capturing motion data from monocular cameras. In this chapter, we present a review of the literature starting with human pose estimation from image to motion reconstruction from video.

2.1 Human Pose Estimation

Human pose estimation (HPE) consists in estimating a representation of the posture of the human body from a monocular image. In computer vision, three models of representation of posture are used in related studies (see Figure 2.1):

— *Skeleton-based model* where the human body is associated to its skeleton-like form. This model represents the human body skeletal structure as a set of joint locations and the corresponding limb orientations. It can also be described as a graph with joints as vertices and limbs as edges connecting the joints within the skeletal structure. Note that the Cartesian coordinates of the joint locations can be defined in 2D or 3D space.

— *Shape-based model* which is a contour-like representation of the human body. It contains width and contour information of the body limbs and torso and is used only in 2D pose estimation.

— *Volume-based model* in which the human body shape and pose is represented with

geometric shapes or meshes. Earlier representation used cylinders and conics for modeling the body parts. Now, most representations use meshes, containing information on the mesh vertices.

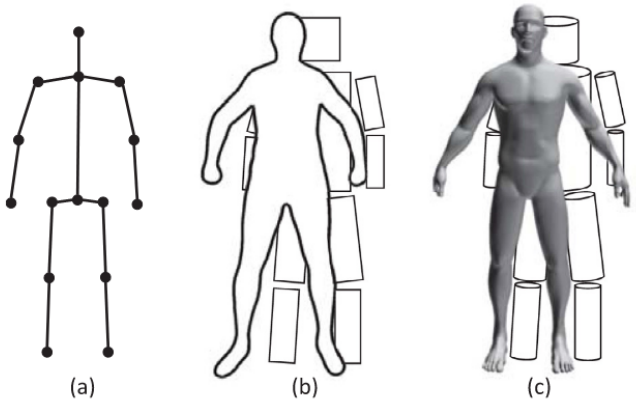

Figure 2.1 – Models of human body representation. (a) Skeleton-based model. (b) Shape-based model. (c) Volume-based model. Source: Chen et al. [19]

In this thesis, we are interested in the skeleton-based model, which is the most used among the three models.

Two branches of studies stem from the HPE task. When remaining in the scope of the image where the estimated data are the 2D pixels coordinates of the skeletal joints, we talk about 2D HPE. When the scope is extended to 3D space, the task becomes a 3D HPE problem.

2.1.1 2D Human Pose Estimation

Estimating human pose started in 2D scope with the objective of detecting 2D joint coordinates of skeletal representations of people in an image. This problem is divided into two contexts: single-person context for image with a single person in it, or multi-person context for images with many people.

Single-Person Pose Estimation

Deep learning (DL) solutions for single-person context are made using either a direct regression approach from the image or using an intermediate heat map prediction (see Figure 2.2). Direct regression approaches use a deep neural network (DNN) to learn the

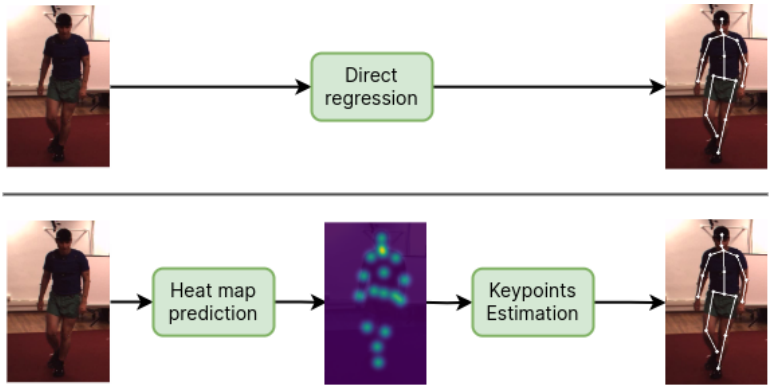

Figure 2.2 – Single-person pose estimation pipelines. Top: Direct regression approach. Bottom: Heat map based approach.

2D joint locations from the image. As an example, Toshev and Szegedy [117] proposed a cascaded DNN regressor to predict the 2D pose representation directly.

Other approaches firstly determine heat map representations of the joints from images and regress them to get the 2D joint locations. In this category, Newell et al. [90] proposed a multi-stage model to learn heat maps through supervised learning, then estimate 2D joint locations.

Multi-Person Pose Estimation

In the multi-person context, the difficulty increases compared to single-person cases. Firstly, there are no prior knowledge of how many people there are in the image. Also, more people means more features to extract from the image. Up to date, there are two methods used in this context, namely *top-down approaches* and *bottom-up approaches*.

Top-down approaches. These approaches detect and locate people in the image and then proceed with the estimation of skeletal postures for each of them. Solutions start with extracting cropped images of each person present with bounding boxes. Each cropped image is then sent to a neural network designed for single-person pose estimation to calculate the 2D joint coordinates. Finally the 2D joint coordinates are brought back to the original image plan using the location of the bounding box. Solutions such as AlphaPose [35], CPN [18] and HRNET [110] are great references to top-down approaches.

These approaches are often considered to be time-consuming since the pose estimation is made for one person at a time. The more people in the image, the longer the estimation time.

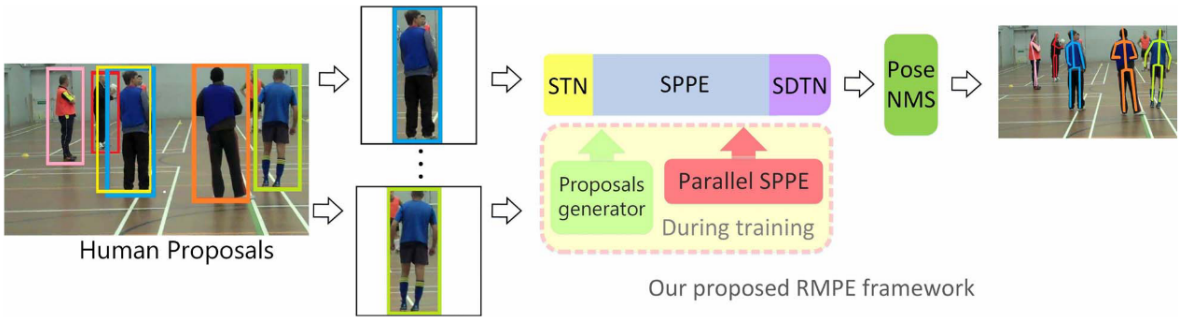

Figure 2.3 – Pipeline of AlphaPose [35] as an example of top-down approach.

Bottom-up approaches. These approaches identify all the human body parts in an image through a first operation. Then, they proceed to reassembling them to reconstruct the different skeletal postures of people in the image. An earlier example of bottom-up approach is DeepCut [97]. This approach proceed in three steps which are 1) detecting the body parts in the image; 2) labeling the different body parts as "leg", "torso" and so on; and finally 3) assembling the body parts belonging to the same person. Among the bottom-up approaches, OpenPose [13] is a well-known and very used solution. In this method, they first detect body parts with a representation called *Part Affinity Fields* and then use a matching algorithm to reassemble the different skeletal postures.

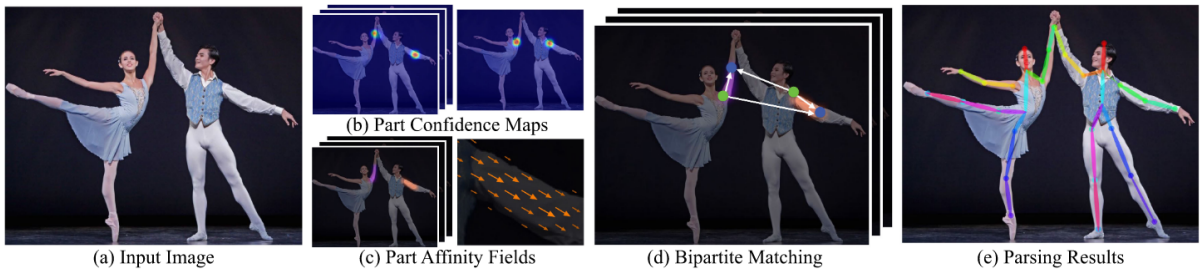

Figure 2.4 – Pipeline of OpenPose [13] as an example of bottom-up approach.

Bottom-up approaches are generally faster than top-down approaches since the estimation is made simultaneously for all people in the image. However, solutions of this category have to be careful during the matching process to avoid mixing body parts of different persons.

Each approach has its pros and cons and the choice is usually made depending of the end purpose. Bottom-up approaches are effective for real-time use, otherwise top-down approaches are usually preferred.

2.1.2 3D Human Pose Estimation

In this branch, the posture representation is estimated in 3D space. This task is more challenging because of depth information which is difficult to deduce from 2D information. For the remainder of this chapter, we consider the single-person context of the problem.

There are two main methodologies to address 3D HPE, either estimation from images or estimation from 2D pose representations obtained before hand with a 2D HPE solution.

Direct estimation from image

This first method consists in processing images to directly estimate position coordinates of the skeleton joints in 3D space, thus, estimating the skeletal posture. Solutions of this group extract features within the image and regress them into 3D position coordinates of the different joints. Various features can be extracted such as heat maps of possible joint locations, depth information hypothesis, etc. Figure 2.5 summarizes the pipeline of this method.

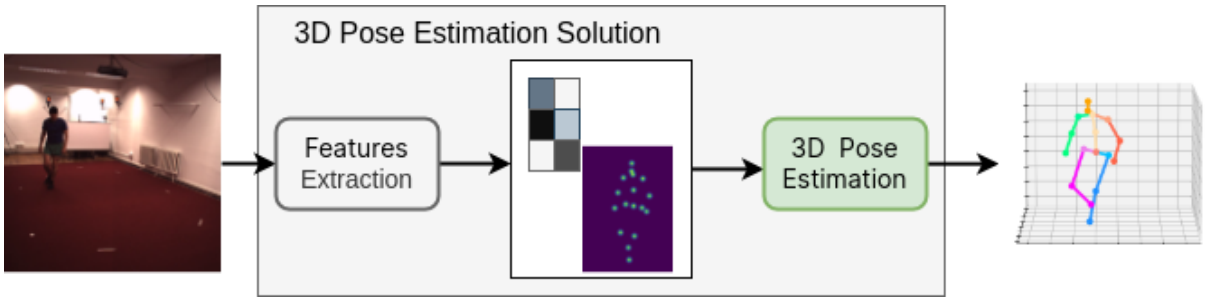

Figure 2.5 – Pose estimation through features extraction from images.

Using this method, Habibie et al. [43] chose to extract 2D heat map representations of the skeletal joints and additional 3D pose information (depth information for example) from image, then integrate them to estimate the 3D joints coordinates. Wei et al. [129] used a framework to generate heat maps and bone maps in order to extract 2D pose hypotheses. They then used a pose regression or a selection-based algorithm on these hypotheses to compute the final 3D pose. Sun et al. [112] propose a solution that extract 3D heat maps with depth information from the image. The 3D heat maps are then regressed into 3D position coordinates for each joint using an integral function, thus leading to the 3D pose representation.

Estimation from 2D pose representation

This second method is conceived on top of 2D HPE algorithms. Approaches of this category propose neural networks that transform 2D poses into 3D poses. This requires to first estimate 2D joint locations from the images using existing solutions such as CPN [18], OpenPose [13]. The neural network then uses the vector of 2D joint locations previously obtained as input data to estimate the vector of 3D position coordinates (representing the 3D skeletal posture) as shown in Figure 2.6.

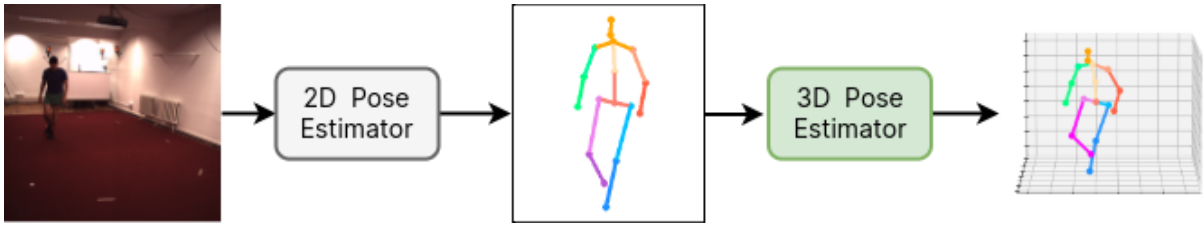

Figure 2.6 – 3D Pose estimation using 2D pose estimation solutions.

Martinez et al. [75] proposed an approach using consecutive linear layers to perform a 2D-to-3D regression of joint positions. Combining a 2D-to-3D pose regression and a 3D-to-2D pose re-projection modules, Biswas et al. [9] used re-projection error minimization as a constraint to predict the 3d locations of body joints. Chen et al. [16] presented an unsupervised algorithm that lifts 2D pose representations to 3D. They showed that adding random 2D projections and an adversarial network allows the training process to be self-supervised using geometric consistency. Shimada et al. [105] decided to first estimate 3D pose from 2D joint locations, and then they used foot contact prediction and physics-based pose optimization to make the estimated pose more realistic. Zou et al. [148] and Zhao et al. [138] opted for a representation of 2D skeletal posture as a graph and used a Graph Convolutional Networks to estimate the 3D skeletal posture from it. Azizi et al. [7] proposed a solution that encodes transformations between joints using the Möbius Transformation and uses a new light Spectral GCN to achieve state-of-the-art results. All these approaches focused on estimating 3D poses with high accuracy and achieved great results in 3D HPE.

Through the various work achieved in HPE, the idea of reconstructing motion from video has taken place, leading to the 3D motion reconstruction task.

2.2 Human Motion Reconstruction

Human motion reconstruction (HMR) involves capturing and reconstructing motion from video data. This field of research has started to develop thanks to the progress made in HPE and deep learning (DL) techniques. Before reviewing existing methods for 3D motion reconstruction, we present different numerical representations of motion used for adaptability to deep learning methods requirement.

2.2.1 Human Motion Representation for Deep Learning Methods

Using deep learning methods to reconstruct human motion, requires a numerical representation of it. Related studies used various representations defined from the skeleton-based model of the human body.

Positional Representation

In positional representation, a posture is described by the 3D position coordinates of the skeletal joints. Each posture is encoded as a vector of joint positions. The posture can be visualized in drawing as shown in Figure 2.7. Human motion is considered as a sequence of postures in chronological order. Thus, it is encoded as a vector of vectors corresponding to the position coordinates of the joints. Most approaches choose this representation as it is the simplest and most easy to use [96, 140, 135].

This representation is the most compatible with previous studies on 3D human pose estimation where the posture is represented by a vector of position coordinates of the skeletal joints.

The positional representation does not explicitly take into account the interdependence between the time and space axes. In fact, the only link between them is related to the fact that the postures are ordered in a chronological order. To encode the correlation between the time and space axes, this representation can be modified to represent the motion as a spatiotemporal graph. Each skeletal joint is a node of the graph. The edges of the graph are created by pairing the joints of each posture at each time instant to form the skeleton. Additional edges to represent the temporal connection between the postures are made, connecting similar joints through time. This representation was used in many approaches [125, 137].

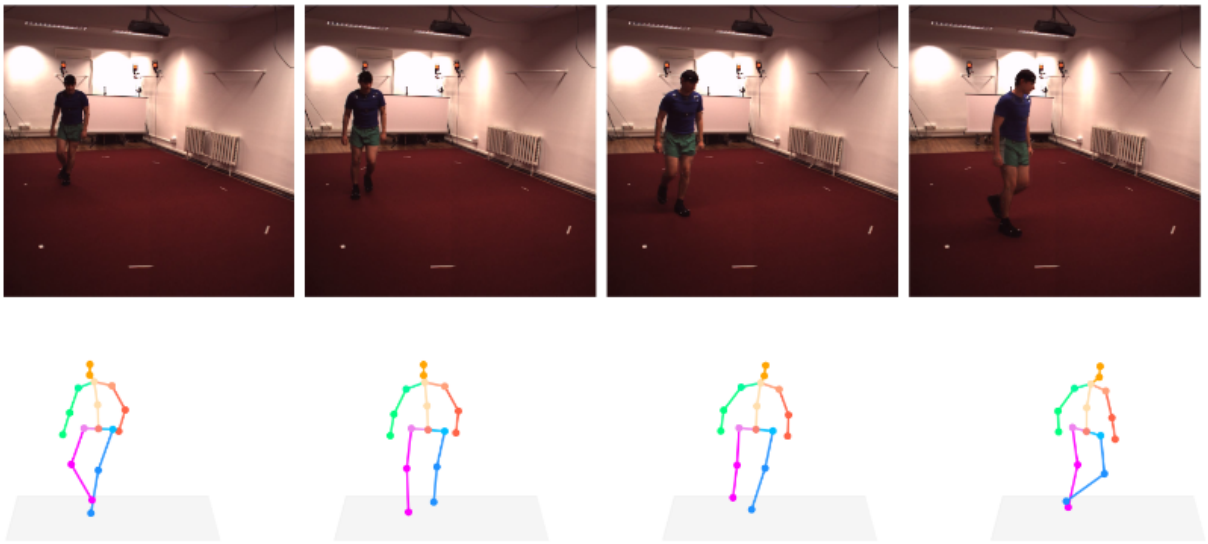

Figure 2.7 – Representation of human motion as a sequence of 3D skeletal poses. Top: sequence of images from video. Bottom: sequence of 3D skeletal poses.

Angular Representation

This representation is inspired by the kinematic animation of articulated bodies. Considering that the skeleton of the human body is an articulated body, that is bones connected with articulations (skeletal joints), two types of information are encoded as illustrated in Figure 2.8. Static features that parameterize the skeleton such as bone lengths and dynamic data that defines the motion of the skeleton. Dynamic data represents the changes over time of rotation angles of each skeletal joint and are sequentially encoded as a vector of vectors. The motion is encoded as the set of both the static and dynamic data.

An advantage of this representation is that it ensures that bone lengths remain constant over time. A few DL approaches for motion reconstruction [95, 105] use this representation. Using this representation sometimes create discontinuities when regressing joint rotations as Euler angles, therefore Pavllo et al. [95] proposed to represent the rotations with quaternions. Using forward kinematics algorithms [4] on this representation, it is possible to compute the positional representation thus taking into account joint position errors during the regression. Due to the characteristics of 3D rotations a specific representation may be required for neural networks as proposed by Zhou et al. [146].

Positional representation is usually preferred to angular representation when deep

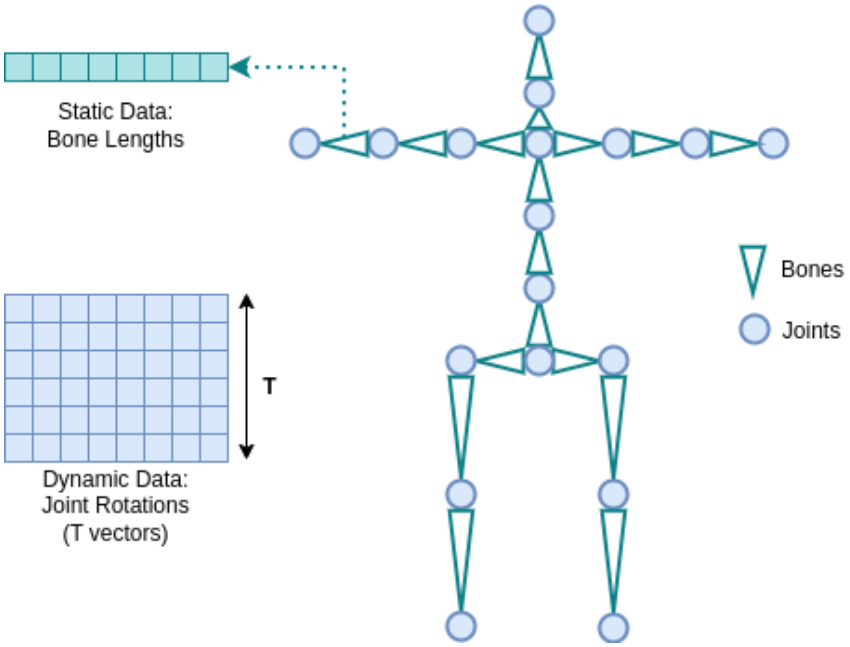

Figure 2.8 – Example of angular representation of human motion. For a motion of length T, the dynamic data consist of T vectors containing rotation angles of each joint.

learning methods are involved for ease of use.

2.2.2 Deep Learning Methods for Motion Reconstruction

Deep learning methods for motion reconstruction include methods that consider not only the spatial accuracy of the joint positions at each pose but also the temporal consistency between the skeletal postures in the sequence representing the motion. This group consists of solutions designed to estimate a sequence of 3D human pose from a sequence of images or a sequence of 2D poses. Because the estimation is made from a sequence, the estimated 3D poses have a better temporal consistency. We divided these solutions into two categories presented below.

The first category include methods that proposed deep neural networks for sequence-to-pose or sequence-to-sequence estimation. In sequence-to-pose estimation, DNNs process multiple consecutive frames in the sequence to estimate the posture of the central frame, while in sequence-to-sequence estimation, models process the whole sequence to yield the estimated sequence of 3D skeletal postures. In this category, Hossain et al.[47] used an LSTM-based approach with encoder-decoder units. The decoder includes the estimation at time $t-1$ to estimate the pose at time t, thus applying a temporal smoothness constraint

to the reconstruction process. Zheng et al.[140] used spatiotemporal transformers to learn the motion structure at each frame and between frames. To further improve temporal coherence, some approaches use an additional loss based on temporal features. Cai et al. [12] represented the pose sequence as a spatiotemporal graph and learn the spatial and temporal relationships between joints by means of graph convolution, using a loss function that combines the positional derivative and positional loss. Wang et al. [125] proposed what they call a motion loss, that is a loss function consisting of distances calculated from the projection of the sequence of poses into a motion space. The latter is the encoded representation of the pose sequence obtained by concatenating pairwise cross-product vectors between the coordinate vectors of the same joints over time with various intervals. This motion loss is then added to the joint position loss function to train a graph convolution network. Zhang et al. [135] used *Transformers* at both spatial and temporal levels with a weighted per joint position loss and a temporal coherence loss. These approaches not only improve the temporal consistency of the reconstructed motion, but also its spatial accuracy. However, they do not achieve consistency at the level of skeletal bone lengths.

The second category involves methods that, inspired by computer animation, used the angular representation of motion. Pavllo et al. [95] represented rotations with quaternions and defined a loss function that performs forward kinematics to penalize absolute position errors instead of angle errors. Shi et al. [102] chose to estimate both a skeletal model and the temporal sequence of joint rotations through two parallel convolutional neural networks. They then applied a forward kinematics (FK) algorithm to obtain the sequence of 3D poses of the motion, enabling them to achieve skeletal consistency.

It should be noted that it is more difficult for neural networks to estimate joint rotations than joint positions because different rotations can produce the same poses. As a matter of fact, most of these solutions used while training their model, loss function on joint positions after computing the positional representation with FK methods.

2.2.3 Temporal Loss Function

The loss function generally used to train neural network in 3D human pose estimation is the *joint position loss*. It is the average distance error on the joint positions between the estimated pose and the ground truth. Most solutions use this loss function in their learning process. However, for a motion reconstruction task, it is lacking because it focus on spatial accuracy alone and overlook the temporal axis of the motion. As a solution to

this issue, some researchers propose new loss functions based on temporal characteristics of the motion. Among them, some choose to calculate the loss function by using the first derivative, that is the velocity [28]. Wang et al. [125] propose a loss function, called *Motion Loss*, computed from an encoded representation of motion. They project the predicted and ground truth sequences of poses into the encoding space and compute the difference between the two encoded information. This difference evaluates the temporal quality of the reconstructed motion compared to the ground truth.

These solutions approaches show that coupling temporal loss function with the joint position loss improves the temporal quality of reconstructed motions.

2.3 Summary and Discussions

From 2D pose estimation to 3D pose estimation to finally 3D motion reconstruction, this chapter reviews various studies and experiments conducted using deep learning methods.

With 2D pose estimation, the skeletal posture of people are visualized in images. The results can be achieved using top-down approaches which first detect people in an image and then compute the skeletal joint locations for each of them. Another method grouping bottom-up approaches, is to estimate all body parts in the image and reassemble them to form each people posture.

Adding depth information, 3D pose estimation gives a 3D view of the skeletal posture. Pose representations are calculated either through features extraction from images then regression to calculate 3D joint coordinates or through 2D-to-3D lifting operation of 2D pose representation to integrate depth information.

Finally, motion is represented as a sequence of 3D postures which include the time axis, thus enabling its reconstruction from video. To ensure the temporal coherence of the reconstructed motion, various means such as specialized deep learning methods for temporal data processing, and temporal loss function are used.

Motion reconstruction using deep learning techniques can be used to produce motion data from video. Even though there are improvements in the quality of the reconstructed motion, there are still some improvements needed to effectively use these solutions for high precision motion acquisition. In chapter 6, we will present a system to improve the quality of motion reconstructed through these methods.

Chapter 3

Human Motion Data Collection

Contents

Since the 2000s, technologies based on motion capture, *MoCap*, have made it possible to capture human movement more accurately. Thanks to this development, various human motion databases have been recorded for research purposes. The recorded data can be used in data-driven motion synthesis models in the field of Computer Animation. More recently, the development of deep learning techniques in Computer Vision, led to the generation of new databases, featuring real videos, *MoCap*, and videos constructed from *MoCap*. Section 3.1 lists existing motion databases while section 3.2 deals with disability movement databases.

3.1 Existing Human Motion Databases

We divide the existing motion databases into two groups. Databases obtained through traditional *MoCap* technologies and databases recorded through videos.

3.1.1 Motion Capture Databases

Motion Capture or *MoCap* refers to some techniques for digitally recording motion data from a person's movements. The acquisition is realized using passive or active systems. Passive systems use infrared cameras to record the position and displacement of reflective markers placed on an actor's body. Active systems use different types of sensors, whether inertial, magnetic or mechanical to compute the motions of the actor. We introduce in details some of the databases recorded with these technologies.

One of the first and large *MoCap* database was proposed by the Canergie Mellon University [26]. The **CMU Graphics Lab Motion Capture Database** is built with a variety of categories of motion. It is a free of use database often used in computer animation. It contains motions of human interaction, interaction with the environment, human behaviors, human locomotion, and also physical activities and sports. The recorded motions involve human interaction, interaction with the environment, locomotion (running, walking, etc.), physical activities and sports (basketball, dance, soccer, etc.), for a total of 2,605 clips.

The **Mocap Database HDM05** [86] supplies free motion capture data mostly for motion recognition purpose. For this database, they used a system based on optical marker-based technology, with the actor equipped with a set of 40-50 retro-reflective markers attached to a suit. All recordings were performed at a sampling rate of 120 Hz. The motion sequences were executed several times by 5 non-professional actors. The sequences were manually cut into motion clips arranged into approximately 100 classes of motion. This amounts to roughly 1,500 motion clips with 50 minutes of motion data.

For specific research context, *MoCap* data are extended to generate databases with various contents. These databases provide in addition to motion capture data, other types of data such as: synchronized videos, 3D model of human body data based on SMPL [70](Skinned Multi-Person Linear) Model or its extended version SMPLX [94] (see Figure 3.1), etc.

The **HumanEva** database [106] is an example of such database. Its first version, **HumanEva-I** contains 7 calibrated video sequences (4 grayscale and 3 color) for synchronization with 3D body poses obtained from a *MoCap* system. Only a part of the *MoCap* data are synchronized with video data using a synchronization software. The database contains 4 subjects performing 6 common actions (e.g. walking, jogging, gesturing, etc.). The second version, **HumanEva-II** records only 2 subjects performing an extended sequence of actions they called *Combo*. The sequence starts with walking along an elliptical path, then continues on to jogging in the same direction and concludes with the subject alternatively balancing on each of the two feet.

Ionescu et al. [52] propose the **Human3.6M** database. It is the most used database in 3D human pose estimation task. This database of 3D human poses was captured indoor with a marker-based motion capture system. The data consists of 15 scenarios of daily actions executed by 11 professional actors (6 males and 5 females) and recorded by 4 digital video cameras and 10 motion cameras. The performed motions involve giving directions,

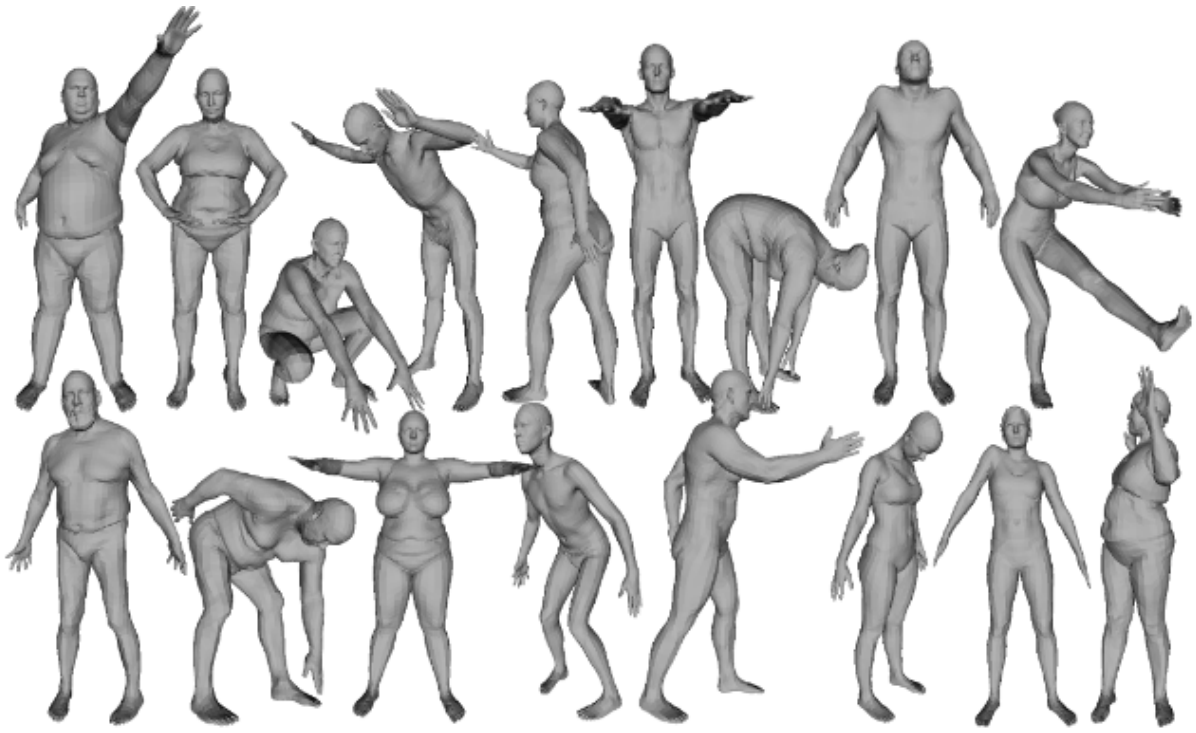

Figure 3.1 – Example of SMPL model taken from the paper of Lopez et al. [70].

discussing, eating, sitting, making purchases, taking photo, smoking, etc. It contains 3.6 million video frames annotated with 3D positions and rotations (ground truth) acquired at a frequency of 50 frames per second. It also provides 3D scanner data of the different actors.

There are other more specific databases oriented to specific motions recorded in a similar environment as the one used for **Human3.6m** [52], namely **Fit3D** [36], **HumanSC3D** [37], **CHI3D** [38].

Fit3D [36] contains 611 multi-view motion sequences involving 47 fitness exercises such as warm-ups, dumbbell exercises, barbell exercises, and equipment-free exercises. The motions were performed by 11 actors, including 1 trainer and 10 trainees. The recordings amount to a total of 2,964,236 video frames with the corresponding 3D skeletal poses.

HumanSC3D [37] database was generated to capture motions with various self-contact occurrences like crossing legs, touching one's head with one's hands. The recordings include 172 motions performed by 6 subjects (3 men and 3 women between 20 and 30 years old with various fitness levels and body shapes). The recorded motions are divided into 116 motions while standing, 20 while sitting on the floor and 36 while sitting on

or standing next to a chair, summing up to a total of 1,246,487 video frames with the associated 3D skeletal poses.

CHI3D [38] is a database related to motions with interactions (handshakes, hugs, holding hands, etc.). Each recording is realized simultaneously by a pair of subjects (5 pairs in total), with only one of the subject being tracked with the marker-based motion capture system while the other is only tracked by the RGB cameras. 631 interaction sequences were recorded for a total of 728,664 ground truth 3D skeletons.

These 4 databases are part of a series recorded in the same environment presented in Figure 3.2.

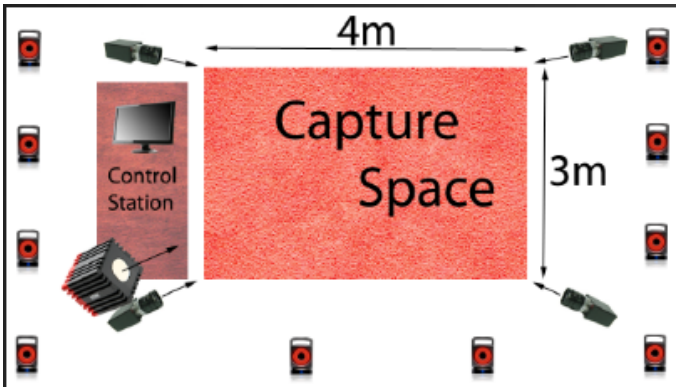

Figure 3.2 – Laboratory setup for marker-based motion capture. There are 10 motion cameras, 4 digital RGB cameras.

AMASS [73] is a large database of human motion, unifying 15 optical marker-based *MoCap* databases such as **CMU** [26], **HDM05** [86, 85], **KIT** [74]. Using the *MoCap* data, motions with 3D human representation in the format SMPL [70] were generated as shown in Figure 3.3. The resulting database amounts to 11,265 motions, more than 40 hours of recordings data from 344 subjects.

We can also find *MoCap* databases designed for more specific contexts such as a corpus for theatrical expressive gestures proposed by Carreno-Medrano et al. [15]. The same research team also proposed French sign language (LSF) databases for data-driven synthesis applied to signing avatars. These databases are described in [32, 64, 41, 88] and their use for synthesis in [45, 42, 89].

There are more motion databases designed for specific purposes in specific contexts.

Figure 3.3 – SMPL body shape generation in AMASS database [73].

3.1.2 Motion Databases Acquired through Video

Video based acquired databases are obtained through filming subjects with one or multiple camera. Motion data are then computed from video data through various techniques. This type of acquisition is usually referred to as marker-less motion capture.

In this regard, **CMU Panoptic** [56] is a typical example of motion database obtained in a studio with 480 synchronized cameras. The database currently contains 5.5 hours of recording divided into 65 sequences. The sequences are made of scenes capturing multiple people interacting with each other.

MPI-INF-3DHP [76] uses a commercial marker-less motion capture system to collect data. The recordings are made in a multi-camera studio of 14 cameras. The database contains 8 activities set performed by 8 actors, 4 men and 4 women. The activities range from walking and sitting to complex exercise poses and dynamic actions.

The database **3DPW** [122] was captured in the wild through a method they called Video Inertial Poser (VIP). The method records motion data using a single moving camera coupled with 6 to 17 Inertial Measurement Units (IMUs) attached to the body limbs. The database contains 60 sequences at 60 Hz for more than 51,000 frames. A total of 7 actors were recorded performing various activities such as shopping, doing sports, hugging, discussing, capturing selfies, riding bus, playing guitar, relaxing.

Motion acquisition using video-based systems is relatively cheaper and easier to use, especially acquisition through a single camera. Although the accuracy of the algorithms used to compute motion data from video is constantly improving, they remain inferior to extremely precise and accurate *MoCap* systems.

3.2 Motion Disability Databases

After browsing the existing motion databases, we found that there are no existing database of motions in physical disability situations. This is mainly due to the difficulty of using traditional motion capture systems on people with disabilities, particularly those with motor impairments who use wheelchairs. Another reason is the belief that what can be achieved with existing motion databases can also be achieved for motor disability cases. Therefore, no attempt has been made to generate such a database.

However, that logic does not apply to some tasks, such as fine motion and behavior analysis. Furthermore, the reconstruction of these constrained movements from videos requires the artificial intelligence (AI) model to have previously learned from such data.

As a result, we decided to create a new corpus of motion in motor disability situations that can be used later in many applications. In our particular case, we want to experiment motion reconstruction from video.

3.3 Summary and Discussions

This chapter reviews existing motion databases acquired through various recording systems. Most of the databases are obtained using traditional *MoCap* systems. Some of these databases were recorded while synchronizing *MoCap* systems with monocular cameras in order to jointly obtain a video of the recorded movement, thus, generating videos data coupled with 3D body poses for task such as AI models training.

There are also a few motion databases recorded using systems with one or more cameras for multiple views to conduct marker-less motion capture. Although less precise than the traditional approach, these systems are improving and are constantly researched for their ease of use.

Finally, we discovered that there are no available *MoCap* database recorded in disability case situations. We believe this is due to the difficulty of capture such motion using traditional systems. Also, actual video-based systems for capturing motion are not sufficiently accurate to use them for task involving motion data in disability cases.

In Chapter 7, we will present a new database of motions in motor disability situations called *Handi-Motion*. This database was generated to design an AI model for marker-less motion capture based on motion reconstruction from video. Considering the difficulty of capturing motion in these situations with the traditional approach, we built this database

by generating 2D videos using 3D captured motions. Indeed, we chose to first capture a few motion with a *MoCap* system, then, used these motion to animate virtual characters and record videos data. With this original approach, we expanded the database using several recording cameras with different viewpoints and varying 3D models of virtual characters.

Part II

Motion Processing and Deep Learning

Chapter 4

Human Motion Modeling as a 3D+t Graph and Discrete Laplacian Operator

Contents

4.1 Introduction

In this chapter, we focus on the numerical representation of motion in order to reconstruct it from video through deep learning techniques. We recall that the human body is simplified to a skeleton-based model. This model represents the human body as an articulated body made of rigid segments (bones) connected by joints. It is a hierarchical configuration of the skeleton that defines the body posture.

In a kinematics point of view, we consider human motion as sequence of posture changes of the skeleton over time. A pose or posture is the state of the skeleton at a given time or frame, described by the position and orientation of each joint. Therefore, the posture s_i of the skeleton at time i is given by:

$$s_i = \{(p_i^1, q_i^1), (p_i^1, q_i^1), \ldots, (p_i^n, q_i^n)\} \tag{4.1}$$

where n is the number of skeletal joints in the skeleton and (p_i^j, q_i^j) are respectively the position and the orientation of joint j at time i with $1 \leq j \leq n$. A motion is then noted

as $M = \{s_1, s_2, \ldots, s_t\}$ with t the total number of postures in the sequence.

Unlike applications in computer animation where the orientation is often needed, in Computer Vision, this information can be overlooked as it is possible to visualize the posture using only the position. Therefore, to facilitate processing, a posture s_i can be simplified to a configuration given by $s_i = \{p_i^1, p_i^2, \ldots, p_i^n\}$. In the remainder of this work, we will use this configuration.

The representation of the motion as a sequence of postures (positions of joints) overlook the temporal connection between the different postures. This issue can be solved by representing the motion as a graph in 3D Euclidean space and guided by distance to include the time and space connections between the skeletal joints.

More than that, this representation allows us to apply Laplacian properties of graph on the motion to study local deformation of joints. With these properties, we can encode the temporal coherence using a differential information to express the motion graph using the discrete Laplacian operator. Also, we can introduce constraints to the graph in order to preserve some important information such as bone length, so that the skeleton remains consistent throughout the sequence of poses.

In this chapter, we will first present the discrete Laplacian operator applied to a graph. We will then explain the representation of the motion as a graph. And, finally, we will present the advantages of this representation.

4.2 Discrete 3D Laplacian Operator

The discrete Laplacian operator is often used in computer science to analyze and edit geometric shapes. In this section, we show how the discrete Laplacian operator is applied to a graph in 3D Euclidean geometry setting.

4.2.1 Discrete Laplacian Operator

To describe this operator in the current case, let us consider a graph $G = (V, E)$ where $V = \{\mathrm{v}_1, ..., \mathrm{v}_n\}$ is the set of vertices of the graph in Euclidean geometry and E the set of edges. Let the function $p : V -> \mathbb{R}^3$ that assigns to each vertex v_i, a vector corresponding to its Cartesian coordinates. The discrete Laplacian operator applied to p is such that:

$$\forall i, (\mathcal{L}p)(\mathrm{v}_i) = \frac{1}{d_i} \sum_{v_j \in \mathcal{N}(i)} w_{ij}(p(\mathrm{v}_i) - p(\mathrm{v}_j)) \tag{4.2}$$

with w_{ij} the weight associated to the edge $(\mathrm{v}_i, \mathrm{v}_j)$, $\mathcal{N}(i)$ the set of vertices neighbors to v_i and d_i a positive factor defined for vertex v_i.

4.2.2 Laplacian Coordinates

In this setting, the function p returns the vector of Cartesian coordinates of each vertex of the graph in Euclidean space. We will note $p(\mathrm{v}_i) = \boldsymbol{v}_i = [v_{ix}, v_{iy}, v_{iz}] \in \mathbb{R}$. Using the Laplacian operator, we express the graph with differential information that is a set of differential coordinates $\boldsymbol{\delta_i}$ for each vertex v_i. We note $\Delta = \{\boldsymbol{\delta_i}\}$. The Laplacian coordinates $\boldsymbol{\delta_i} = \mathcal{L}(\mathrm{v}_i)$ of vertex v_i is a 3-dimensional vector $\boldsymbol{\delta_i} = [\delta_{ix}, \delta_{iy}, \delta_{iz}]$ that can be interpreted as the difference between its vector $\boldsymbol{v}_i$ and the mean of all vectors $\boldsymbol{v}_j, \forall v_j \in \mathcal{N}(i)$.

$$\mathcal{L}(\mathrm{v}_i) = \boldsymbol{\delta_i} = \sum_{j \in \mathcal{N}(i)} \mathrm{w}_{ij}(\boldsymbol{v}_i - \boldsymbol{v}_j) \tag{4.3}$$

where w_{ij} is the Laplacian weight applied to edge $(\mathrm{v}_i, \mathrm{v}_j)$. It should be noted that w_{ij} can be used in a non-normalized case $\mathrm{w}_{ij} = w_{ij}$ or in a normalized case, $\mathrm{w}_{ij} = \frac{w_{ij}}{d_i}$ with $d_i = \sum_{k \in \mathcal{N}(i)} w_{ik}$ represented as the sum of the weights of all edges associated with vertex v_i. For the remainder of this chapter, we will consider the non-normalized case.

4.2.3 Matrix Notations

Calculations with the Laplacian operator are generally handled using vectors and matrices. We represent the graph in the Cartesian system by the $(N \times 3)$ matrix P of vectors $\boldsymbol{v_i}$ associated to its vertices:

$$P = \begin{bmatrix} \boldsymbol{v_1} \\ . \\ . \\ \boldsymbol{v_n} \end{bmatrix} \tag{4.4}$$

Let A be the adjacency matrix of the graph and $D = diag(d_1, \ldots, d_n)$ the matrix of degrees. The transformation of the graph from its representation with Cartesian coordinates P into a representation with Laplacian coordinates Δ can be given by $\Delta = LP$ where $L = I - D^{-1}A$. The matrix L is the matrix representation of the Laplacian operator of

the graph defined by the formula:

$$L_{ij} = \begin{cases} \sum_{j \in \mathcal{N}(i)} \mathrm{w}_{ij} & \text{if } i = j \\ -\mathrm{w}_{ij} & \forall (i,j) \in E \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

4.3 Human Motion Representation as a Laplacian Graph

In this section, we propose a representation of the human motion as a graph in order to apply the discrete Laplacian operator on it.

4.3.1 Human Motion Graph

Based on the definition of human motion as the change of skeletal postures over time, we consider two axis: the spatial axis represented by each posture and the temporal axis.

For the spatial axis, the skeleton representation that defines the posture is made of bones and joints. To build a graph from that representation, we consider the joints as the vertices and the bones as the edges as shown in Figure 4.1. The temporal axis is represented

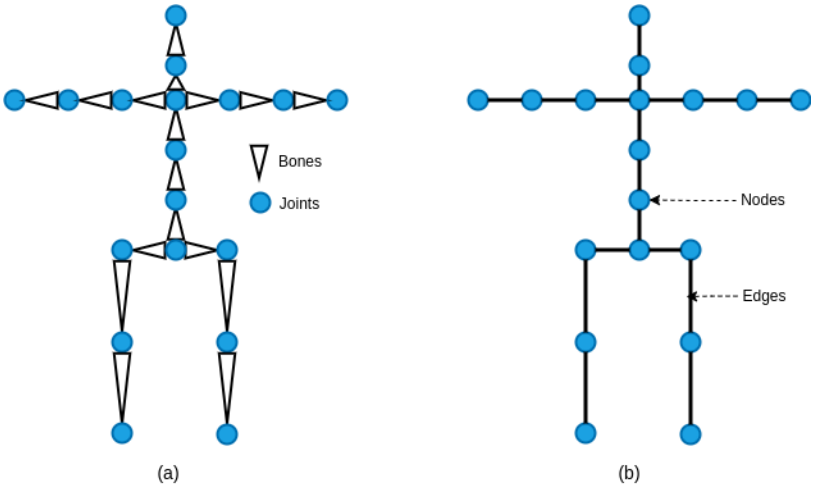

Figure 4.1 – (a) Skeleton as an articulated body. (b) Pose representation as a graph.

as the different skeleton representation of the body at each instant (frame), from the beginning to the end of the motion. More simply, a motion of T frames is considered as

a sequence of T postures $S = (S_1, ..., S_T)$. Thus, the motion can be considered as a large graph made by connecting T graphs and it is called a *3D+t graph*.

In details, each skeleton S_t $(1 \leq t \leq T)$ is a sub-graph $S_t = (V_t, E_t)$ with $v_{i,t}$ the vertex of index i in V_t and E_t the edges. The vertices represent the skeletal joints and the edges the bones of the skeleton. The graph 3D+t is built by connecting the sub-graphs $\{S_t\}$ and is defined by $G = (V, E_S \cup E_T)$, where $V = \{V_t\}_{1 \leq t \leq T}$ is the set of vertices (all skeletal joints from S), $E_S = \{E_t\}$ the set of existing spatial edges for each skeleton S_t, and E_T the set of temporal edges connecting the same joints between adjacent skeletons (in the temporal order of the sequence). For a each vertex $\mathrm{v}_{i,t} \in V_t$, E_T contains all the edges $(\mathrm{v}_{i,t-1}, \mathrm{v}_{i,t})$ and $(\mathrm{v}_{i,t}, \mathrm{v}_{i,t+1})$. Figure 4.2 illustrates such a graph for 3 consecutive frames.

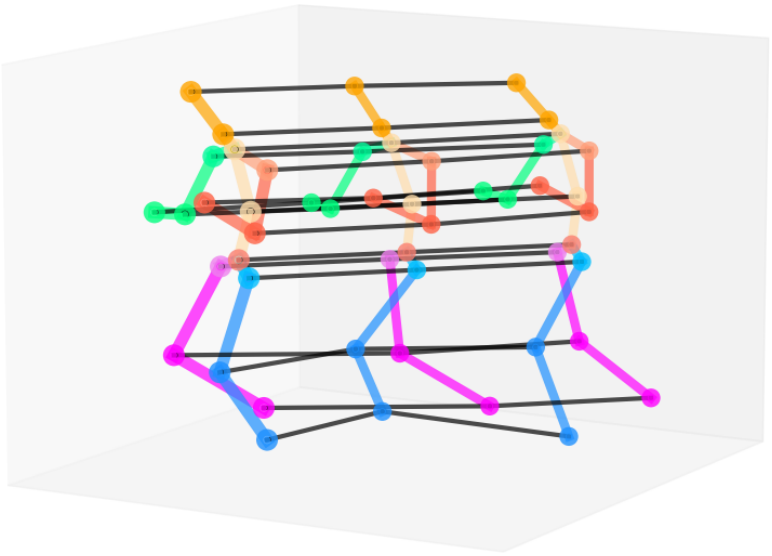

Figure 4.2 – Graph 3D+t for 3 consecutive frames. Each skeleton is represented as a graph forming the set of spatial edges E_S (skeleton bones, in color). Consecutive skeletons are connected through identical joints forming a set of temporal edges E_T (in black).

4.3.2 Application of Discrete Laplacian Operator on Human Motion Graph

With the representation of the motion as a 3D+t graph, it is possible to apply the discrete Laplacian operator on it.

3D+t Discrete Laplacian Operator. As presented by Le Naour et al. [63], while applying the discrete Laplacian operator $\mathcal{L}$ on the 3D+t graph, we compute the Laplacian coordinates vector $\boldsymbol{\delta_{i,t}}$ of the vertex $\mathrm{v}_{i,t}$, represented by the Cartesian coordinates vector $\boldsymbol{v}_{i,t}$, with the formula:

$$\mathcal{L}(\mathrm{v}_{i,t}) = \boldsymbol{\delta_{i,t}} = \mathrm{w}^{-}(\boldsymbol{v}_{i,t} - \boldsymbol{v}_{i,t-1}) + \mathrm{w}^{+}(\boldsymbol{v}_{i,t+1} - \boldsymbol{v}_{i,t}) + \sum_{j \in \mathcal{N}_t(i)} \mathrm{w}_{ij,t}(\boldsymbol{v}_{i,t} - \boldsymbol{v}_{j,t}) \tag{4.6}$$

with w^{-} and w^{+} the weights associated respectively to the temporal edges $(\mathrm{v}_{i,t}, \mathrm{v}_{i,t-1})$ and $(\mathrm{v}_{i,t}, \mathrm{v}_{i,t+1})$, and $\mathrm{w}_{ij,t}$ the weight associated to the spatial edge $(\mathrm{v}_{i,t}, \mathrm{v}_{j,t})$.

Laplacian coordinates are used in this context to represent movement as a set of local deformation of skeletal joints. This encoding provides an accurate modeling of motion that compacts both the spatial and temporal relationships between the joints and can be integrated with deep learning techniques.

4.4 Conclusion

This chapter presents a representation of human motion for motion reconstruction. Using the definition of the motion as a sequence of skeletal postures, we represent each posture as a sub-graph. Then, we build a large graph by connecting the sub-graphs of consecutive postures, thus creating temporal links between them. Between two consecutive sub-graphs, an edge is added to connect vertices representing the same type of joints, i.e. *left ankle* with *left ankle*, *right elbow* with *right elbow* and so on.

We use this representation of human motion in two ways:

— To define a spatiotemporal loss function used to train deep neural network designed to reconstruct motion from videos. This use case is presented in chapter 5

— To build a motion correction system that improves temporal coherence and adjusts skeletal consistency of reconstructed motion from video. We use a neural network that operates on Laplacian coordinates to improve the spatiotemporal accuracy of the reconstruction and use the flexibility of the graph representation to incorporate additional constraints in order to preserve the skeletal structure. The achieved results are presented in chapter 6.

Chapter 5

Motion Reconstruction with Deep Learning Methods

Contents

5.1 Introduction

We have carried out an in-depth comparative study of DL techniques to achieve the best results in reconstructing 3D motion from video. We consider human motion reconstruction (HMR) as an extension of human pose estimation (HPE). As a matter of fact, DL techniques developed for HPE can be used in HMR. Nevertheless, the end goal in HMR is a little different since, on top of the accuracy in the estimated human poses, the temporal coherence between the different poses is also sought. And as a result, we differentiate between DL solutions for HMR and solutions for HPE.

We consider three main factors to distinguish between solutions for HMR and solutions for HPE, based on the settings defined to build the DL models.

The first factor is the formulation of the problem. Building a DL solution begins with precisely formulating the problem by defining the formats of the input and output of the model. In 3D motion reconstruction, the input data are either a sequence of images (video) or a sequence of 2D poses. The output data is a sequence of 3D poses that represents the motion.

The second factor is the architecture of the neural network. It is related to the different DL algorithms that are used to design the neural network such as convolution, multi-layer perceptron, normalization and so on. We will discuss in this chapter the different possibilities regarding DL networks and present at the end our model architecture to compete with state-of-the-art solutions.

The last factor is part of the process to train the neural network, namely the loss function. This function is used to guide the model towards the results we want it to achieve during the training session. In HPE, the focus is generally on the accuracy of the 3D poses which means that the loss function is related to the error in estimating the 3D joints positions. However, in 3D HMR for computer animation, we also want the resulting movements to be consistent over time. To achieve this, the loss function will also incorporate errors related to temporal characteristics. In addition to the loss function based on motion descriptors and custom loss functions borrowed from the literature, we propose an additional loss function, which is a spatiotemporal one, based on the representation of motion as a 3D+t graph (see sub-section 4.3.2).

After designing and training a DL solution, an evaluation session is proposed to determine whether the results produced by the model are satisfactory or not. The metrics used in this session depend on the nature of the problem. In 3D motion reconstruction, metrics related to the spatial accuracy and the temporal consistency of the motion are usually used.

During this study, we carried out two experiments. The first experiment aimed to test the efficiency of a spatiotemporal loss function that we developed, called *Laplacian loss*. In the second experiment, we designed a complete motion reconstruction solution to compete with existing state-of-the-art solutions.

Chapter outline. We will first review the above mentioned factors that characterize DL solutions for 3D HMR in sections 5.2 and 5.3. Then we will present the different evaluation metrics for the evaluation process in section 5.4. Finally, in section 5.5, we will present the results of our two experiments made using two different neural networks. Firstly we will use a convolution-based neural network to evaluate the efficiency of the Laplacian loss function. And secondly, we will propose a variational autoencoder neural network to achieve state-of-the-art results.

5.2 Deep Neural Network

Prototyping a DL solution requires two analytic steps. The first is to formulate the problem we are trying to solve in machine language, i.e. a formulation that can be understood and solved by a neural network. Next, we need to decide which deep learning algorithms to use to build the model architecture. In this section, we will detail these steps for the specific case of HMR from video.

5.2.1 Formulating the Problem

Formulating a DL problem is usually done by determining two parameters. The formulation requires to: i) define the type of data we are using which will determine the input format of the neural network and ii) define the type of data the model should produce, determining the output format of the model. In computer vision, where it is easier to manipulate motion as sequence of postures, the output format is usually a sequence of 3D poses (3D coordinates of the joints of the skeleton). Then, based on the format of the input data, there are two possible formulations:

1. reconstruction from video (sequence of images)
2. reconstruction from a sequence of 2D poses

The literature shows that the advances made in the 2D human pose estimation such as CPN [18], HRNet [110] or OpenPose [13], make the reconstruction from 2D poses a good and simplified choice. Thus, we will proceed with this method in the subsequent study.

Formulation of 3D HMR problem. The task is to learn the transformation from a sequence of 2D poses into a sequence of 3D poses. This formulation implies that to reconstruct the motion, the video must be first processed into a sequence of 2D joint locations which in turn will be sent to the neural network for a transformation into 3D space. A pose (or posture) can be represented as a vector of the coordinates of the skeletal joints. Let us define X the input data and Y the output. X, representing the sequence of 2D poses, is a vector of vectors containing the 2D joint locations. Y, representing the sequence of 3D postures, is a vector of vectors containing the 3D joint locations. We note $Y = \mathcal{F}(X)$ where $\mathcal{F}$ is the function or algorithm that transforms X into Y. Figure 5.1 summarizes the formulation of the task for DL methods.

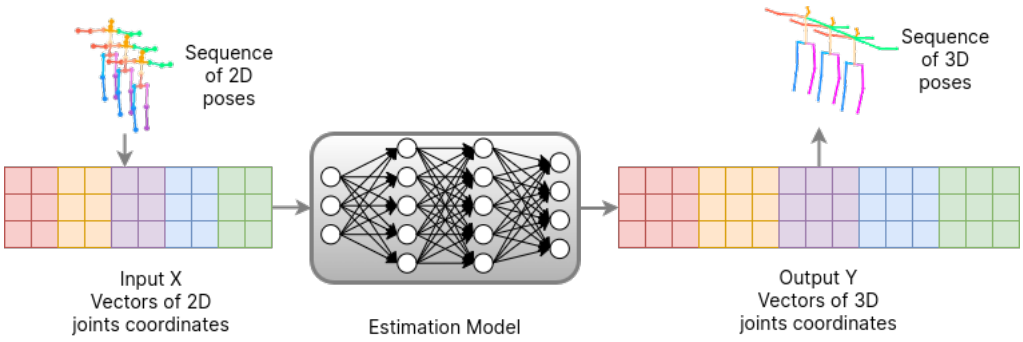

Figure 5.1 – Formulation of the HMR task.

5.2.2 Designing a Neural Network Architecture

After the formulation of the problem, the next step is to design the architecture of the DL solution. The choice of the the DL algorithms depends on the type of method to solve the problem. The literature shows three groups of deep learning DL methods for solving this problem:

1. *Per frame pose estimation.* This approach ignores the time dimension of the problem and focuses on the space. The concept here is to compute the pose in 3D space by learning the depth dimension from a single 2D pose. Most solutions to 3D HPE fit in this category.

2. *Sequence-to-pose estimation.* In this approach, a sequence of 2D poses is used to estimate a single pose in 3D, generally the pose of the central frame. The process is repeated until the whole sequence is finally transformed. The idea behind this method is to make use of the temporal inform between multiple frames to learn a more precise depth information. In this algorithm, the temporal characteristics are partially integrated in the computation process as only one pose is estimated at a time.

3. *Sequence-to-sequence estimation.* The architecture here processes the whole sequence to compute the transformation. This is to ensure that the temporal connection between the 2D poses in the sequence are preserved in the estimated 3D sequence. Among the three methods, this is the only method that fully integrates the time axis in the reconstruction process. Solutions from this category are more suited to the task of 3D motion reconstruction where the temporal consistency between the postures is desired.

The first two methods focus mainly on the spatial accuracy while the last one tries to balance between spatial accuracy and temporal coherence.

In the literature, the proposed solutions use various DL algorithms to design their neural network. Before we design our own, we will browse through some of these algorithms.

Multi-Layer Perceptron

This is one of the first algorithm that introduces the concept of DL. Models are made of multiple fully Connected (also called Dense or Linear) layers. The FC layer

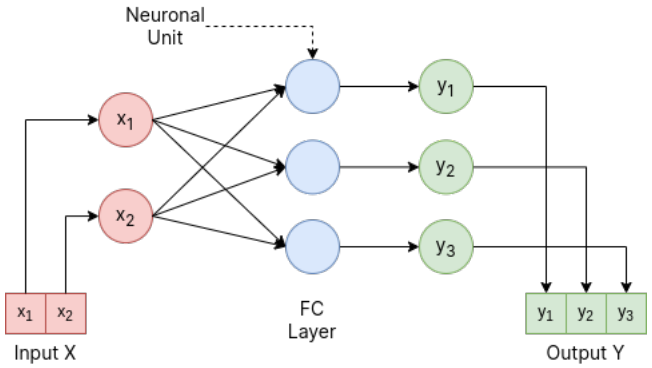

Figure 5.2 – Example of Fully Connected layer.

processes only the features dimension of the input data and thus does not include the time dimension. Applying this algorithm on temporal data implies that the data at each time t is processed independently, and thus, the output sequence does not integrate the temporal characteristics of the input sequence.

Recurrent Neural Network

The recurrent neural network approach is an algorithm specifically made to work with temporal data. It sequentially processes the input data in the chronological order and at each time, it reuses the result of the previous computation. By doing so, the temporal connection between the sequence of data is integrated in the output. This algorithm has the advantage of working with sequences of any size.

There are extensions of this algorithm such as Long Short-Term Memory networks (LSTM) for long time lags information and many others. The most notable disadvantage of this algorithm is that its computation process for large sequences is time-consuming.

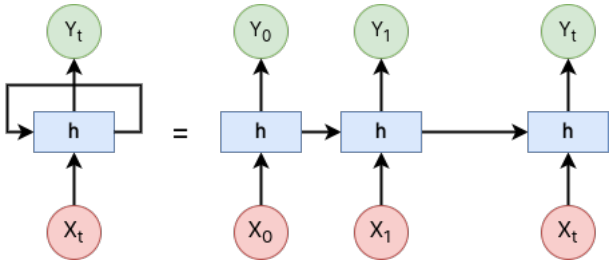

Figure 5.3 – Recurrent Neural Network.

Convolution Neural Network

Convolution Networks are a type of networks built on a convolution algorithm. In the case of temporal data we use 1-D also called temporal convolution. This is the most widely used algorithm that brought a significant breakthrough in the task. The majority of the existing solutions are built upon it. This algorithm is a solution to the time-consuming problem brought by recurrent neural networks.

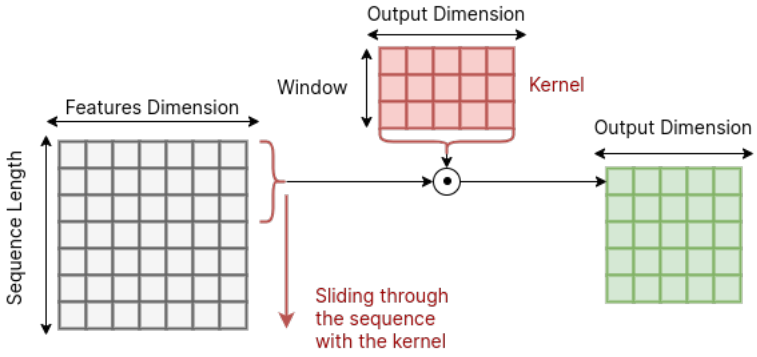

Figure 5.4 – Example of 1-D Convolution for input with multiple features.

Temporal convolution applies filters using a time window (kernels). It takes as input $T_{in} \times C_{in}$ and output $T_{out} \times C_{out}$, where, T_{in} and T_{out} represent respectively the sequence lengths of the input and output, C_{in} the features dimension and C_{out} the output dimension. Moreover, we can use a dilated convolution to apply filters on non-consecutive frames to learn features at different time scales. A fully convolutional neural network has the advantage of not requiring a fixed sequence length and as a result can be easily generalized.

Graph Convolutional Network

Graph convolution is a convolution operation applied to a graph structure. Similar to convolution applied to image where the filter is applied to the neighborhood of each pixel, the operation is applied using the neighborhood of each node of the graph. This operation is realized using the adjacency matrix of the graph that contains the connection between the different nodes.

In 3D motion reconstruction, graph convolution networks can be combined with the representation of the motion as a spatiotemporal graph (see section 4.3).

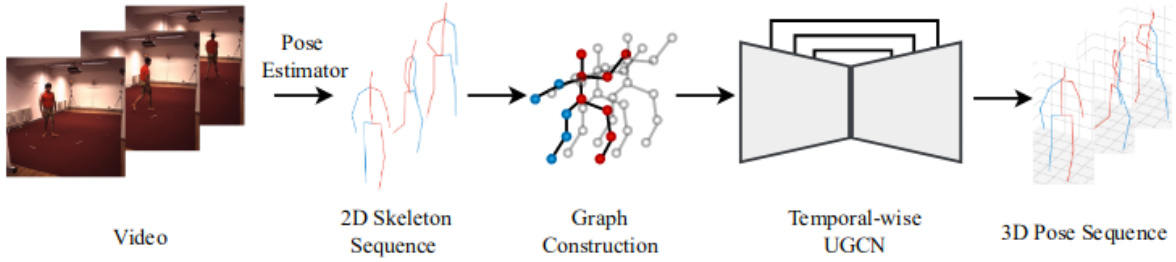

Figure 5.5 – Example of a graph convolution network usage for 3D motion reconstruction. Source: Wang et al. [125]

Transformers Neural Network

The transformer architecture first appeared in Natural Language Processing. With its attention mechanism, it achieved great success in the field. Later, several approaches attempted to apply the principle of this algorithm to the 3D pose estimation task. The results were significant, and today more and more approaches are making full use of it. We have not made an in-depth study of this method but we believe that it is worth mentioning for reference purposes.

5.3 Training a Neural Network

Training a neural network requires to choose a learning pattern (or paradigm) which refers to the settings used to train a deep learning model. It also requires a loss function defined depending on the learning pattern to assist in the training operation. There are

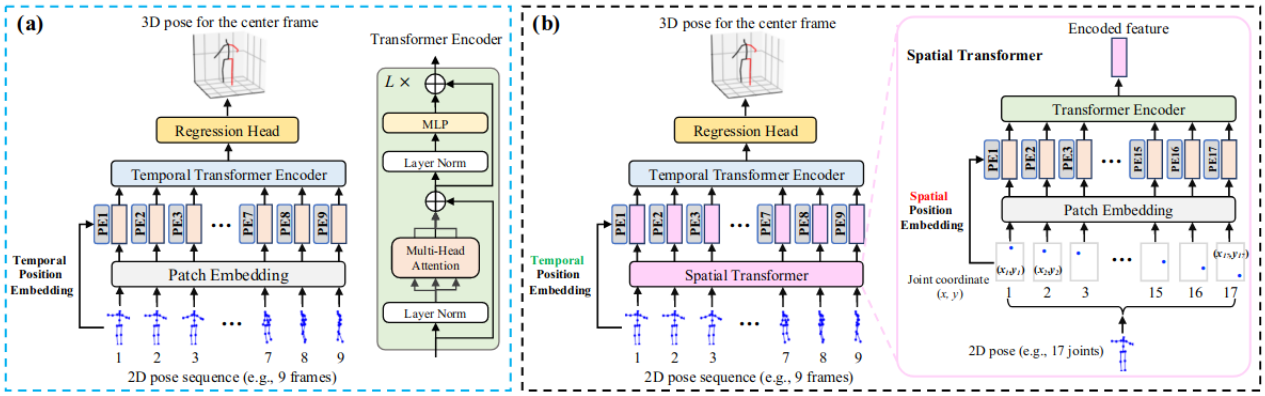

Figure 5.6 – Example of a Transformer based model for human pose estimation. Source: PoseFormer [140]

three learning patterns that can be used to train a neural network in motion reconstruction: supervised, semi-supervised, unsupervised. Among them, we selected the supervised approach which is the most used.

In this section, we will briefly present the different patterns, and later, the loss functions used in the case of supervised learning.

5.3.1 Learning Patterns

In section 5.2, we discussed the architecture of deep learning models for HMR. Once designed, a neural network should be trained on data before its usage. A neural network can be seen as a set of variables or parameters usually referred to as *weights*, used to compute an output from a given input data. Training the model means updating its parameters so that it can compute the right output according to the input. Before choosing a learning paradigm for our experiments, we will briefly review the different paradigms.

Supervised Learning In the supervised learning paradigm, we have at disposition a dataset of labeled data, meaning that for each input, we have a desired output. We can represent the dataset as $\mathcal{D} = (X, Y)$ where the inputs are $X = x_i, 1 \leq i \leq n$ and the targeted values are $Y = y_i, 1 \leq i \leq n$. The learning phase consists of gradually updating the model weights so that for each x_i it can output a value $\hat{y}_i \approx y_i$ (see Figure 5.7).

Unsupervised Learning In the unsupervised learning pattern, the dataset is unlabeled meaning we have the input data X but not targeted output data Y. Updating the model

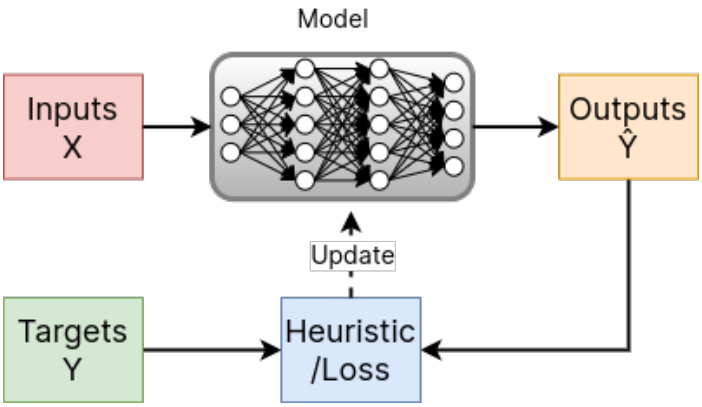

Figure 5.7 – Supervised learning pattern. The heuristic or loss is a quantified difference between the output values and the targeted values. It is computed through a loss function.

is possible through a specific heuristic between the output values inferred by the model and the input values (see Figure 5.8)

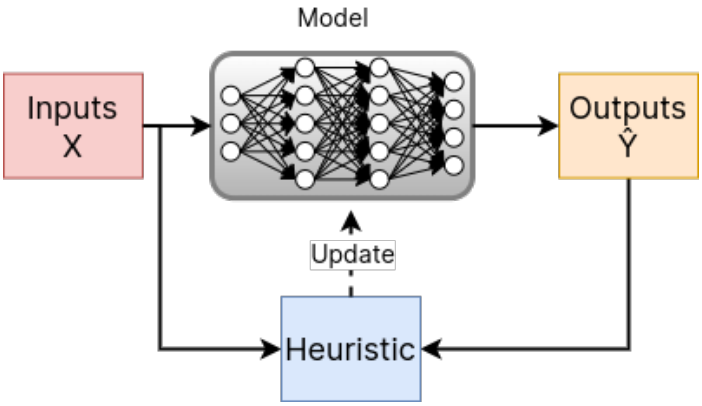

Figure 5.8 – Unsupervised learning pattern. The heuristic is a quantified difference between the output values and the targeted values. It is computed through a loss function.

Semi-supervised Learning The semi-supervised learning is a mix of supervised and unsupervised learning. Indeed, the dataset used in this paradigm is partially labeled i.e., for input data $X = x_i, 1 \leq i \leq n$ we have targeted data $Y = y_i, 1 \leq i \leq m$ with $m < n$. It is also called weak supervision because the number of labeled data is smaller than the number unlabeled data. Using this pattern implies the assumption that there is an underlying relationship between the data. The model is trained to learn this relationship through the labeled data and then transpose it to the unlabeled data.

Selecting a Paradigm Among these different paradigms, we opted in this work for the supervised pattern which is the most used thanks to large-scale annotated dataset such as Human3.6m [52], MPI-INF-3DHP [76], 3DPW [122].

5.3.2 Loss Function

The loss function is a key component in order to efficiently train a deep learning neural network. It is defined depending on the learning pattern chosen to train the model. In the case of supervised training, the function computes an error between the output produced by the model and the desired output. Here we present the various loss functions used in the literature for training neural networks, as well as our own spatiotemporal loss function.

Existing Loss Function

Most studies in 3D human pose estimation, use as loss function the average distance error between the joint positions of the ground truth 3D poses and those of the estimated 3D poses. It is defined as:

$$\mathcal{L}_P = \frac{1}{T} * \frac{1}{J} \sum_{t=1}^{T} \sum_{j=1}^{J} \|\boldsymbol{p}_{t,j} - \overline{\boldsymbol{p}}_{t,j}\|_2 \tag{5.1}$$

where $\boldsymbol{p}_{t,j}$ and $\overline{\boldsymbol{p}}_{t,j}$ are the 3D position vectors of joint j at time t from the estimated poses and the ground truth poses respectively.

This function, if used solely as loss function, works well for single frame pose estimation. But, when working on motion reconstruction, it is limited because it tends to average the joint positions loss over the whole sequence. The less represented poses in the motion can be biased by the more represented ones, affecting the overall motion reconstruction. In the motion reconstruction task, it is preferable to supplement this function with an additional function related to the time axis of the motion. Two temporal loss functions are presented below.

Velocity Loss The first proposition of temporal loss function is the velocity loss function. Based on one of the motion descriptor, the velocity loss refers to the average difference between the velocity vectors of the estimated motion and those of the ground truth.

$$\mathcal{L}_V = \frac{1}{T-1} * \frac{1}{J} \sum_{t=1}^{T-1} \sum_{j=1}^{J} \|\boldsymbol{v}_{t,j} - \overline{\boldsymbol{v}}_{t,j}\|_2 \tag{5.2}$$

where $\mathbf{v}_{t,j}$ and $\overline{\mathbf{v}}_{t,j}$ represent the velocity vectors of joint j at time t, respectively for the ground truth and the estimation.

Motion Loss Function Wang et al. [125] propose a loss function as a distance in motion space. It is based on the encoding motion from a sequence of poses, by computing differential values between the position vectors of the same joint at different time intervals. It can be a subtraction, an inner-product or a cross-product. They encode both the estimated and the ground truth pose sequences. The loss is then computed between the encoded ground truth and the reconstructed poses.

5.3.3 Laplacian Loss Function

Each of the existing loss functions considers either the spatial axis or the temporal axis. As a result, most approaches tend to use a weighted combination of them. We believe that a better solution is a function that implicitly considers both axis at the same time. For that reason, we define a new function that we call *Laplacian Loss*. It considers the local deformation induced by the spatiotemporal graph representation of the motion.

The *Laplacian Loss* computes a difference in the representation of the ground truth motion and the estimated motion in the Laplacian space. As we represent the motion as a graph 3D+t (see Section 4.3.2), characterized by the Laplacian differential coordinates Δ, this loss computes the average error on the differential coordinates of each joints' position between the estimated representation and the ground truth representation. The loss function is defined by the following equation:

$$\mathcal{L}_{\Delta} = \frac{1}{T} * \frac{1}{J} \sum_{t=1}^{T} \sum_{j=1}^{J} \|\boldsymbol{\delta}_{t,j} - \overline{\boldsymbol{\delta}}_{t,j}\| \tag{5.3}$$

where $\boldsymbol{\delta}_{t,j}$ and $\overline{\boldsymbol{\delta}}_{t,j}$ represent the differential Laplacian vectors of joint j at time t from respectively the ground truth and the estimation.

Training a neural network with this loss function enables it to learn the spatiotemporal connections between the joints locations it predicts, thus implicitly taking into account the skeletal structure and the temporal changes of the joints.

5.4 Evaluation Metrics

The numerous studies carried out on the estimation of human pose in 3D have produced a number of standards on how to evaluate and compare proposed solutions. These standards concern evaluation protocols as well as metrics used as reference for quantitative evaluation. We are interested here in the evaluation metrics.

In HPE, the reference metric is a spatial metric linked to the accuracy with which the model estimates the 3D joint positions compared to ground truth. But, in HMR, as the time axis is also considered, temporal metrics related to the temporal features of motion are included in the evaluation.

5.4.1 Spatial Metrics

The most common metrics used in the evaluation process is the Mean Per-Joint Position Error (*MPJPE*) which is related to the spatial accuracy of the reconstruction. It is defined as the average error in estimating the joint positions.

$$MPJPE = \frac{1}{T} * \frac{1}{J} \sum_{t=1}^{T} \sum_{j=1}^{J} \|\boldsymbol{p}_{t,j} - \overline{\boldsymbol{p}}_{t,j}\|_2 \tag{5.4}$$

where $\boldsymbol{p}_{t,j}$ and $\overline{\boldsymbol{p}}_{t,j}$ represent the position vectors of joint j at time t from respectively the ground truth and the estimation.

5.4.2 Temporal Metrics

We call temporal measurements those related to the temporal characteristics of the motion. These metrics are based on kinematic motion descriptors, namely velocity and acceleration. They serve as a reference to evaluate the quality of the reconstructed motion compared to the original one. These metrics are presented below.

Mean Per-Joint Velocity Error, MPJVE This metric is based on the velocity. It is the average distance error between the velocity vectors of the estimation and the ground truth.

$$\boldsymbol{v}_{j,t} = \boldsymbol{p}_{j,t+1} - \boldsymbol{p}_{j,t} \tag{5.5}$$

$$MPJVE = \frac{1}{T-1} * \frac{1}{J} \sum_{t=1}^{T-1} \sum_{j=1}^{J} \|\boldsymbol{v}_{t,j} - \overline{\boldsymbol{v}}_{t,j}\|_2 \tag{5.6}$$

where $\boldsymbol{v}_{t,j}$ and $\overline{\boldsymbol{v}}_{t,j}$ represent the velocity vectors of joint j at time t, respectively from the ground truth and the estimation.

Mean Per-Joint Velocity Error, MPJAccE The MPJAccE is an acceleration-based metric and is computed as the average distance error between the acceleration vectors of the estimation and the ground truth.

$$\boldsymbol{a}_{j,t+1} = \boldsymbol{p}_{j,t+2} - 2 * \boldsymbol{p}_{j,t+1} - \boldsymbol{p}_{j,t} \tag{5.7}$$

$$MPJAccE = \frac{1}{T-2} * \frac{1}{J} \sum_{t=1}^{T-2} \sum_{j=1}^{J} \|\boldsymbol{a}_{t,j} - \overline{\boldsymbol{a}}_{t,j}\|_2 \tag{5.8}$$

where $\boldsymbol{a}_{t,j}$ and $\overline{\boldsymbol{a}}_{t,j}$ represent the acceleration vectors of joint j at time t, respectively from the ground truth and the estimation.

5.5 Experiments and Results

We have previously defined a Laplacian loss function suitable for motion reconstruction. We have chosen to verify the efficiency of this loss function through the following two studies. The first experiment is an ablation study on the Laplacian loss function applied on our own neural network (a simple convolutional model). In the second experiment, we built a more advanced neural network which we trained with the Laplacian loss function and then benchmarked against 3 other state-of-the-art (STAR) models.

5.5.1 Ablation Study on Laplacian Loss Function

The goal of this experiment is to assess the efficiency of the *Laplacian loss* function in training neural network. To do this, we designed a simple deep neural network architecture and trained different versions of it with different loss functions each time. We then analyzed the results and evaluated the most efficient version.

Generic Neural Network CVM-Net

We built this network using temporal convolution algorithms. Figure 5.9 shows the architecture of this model.

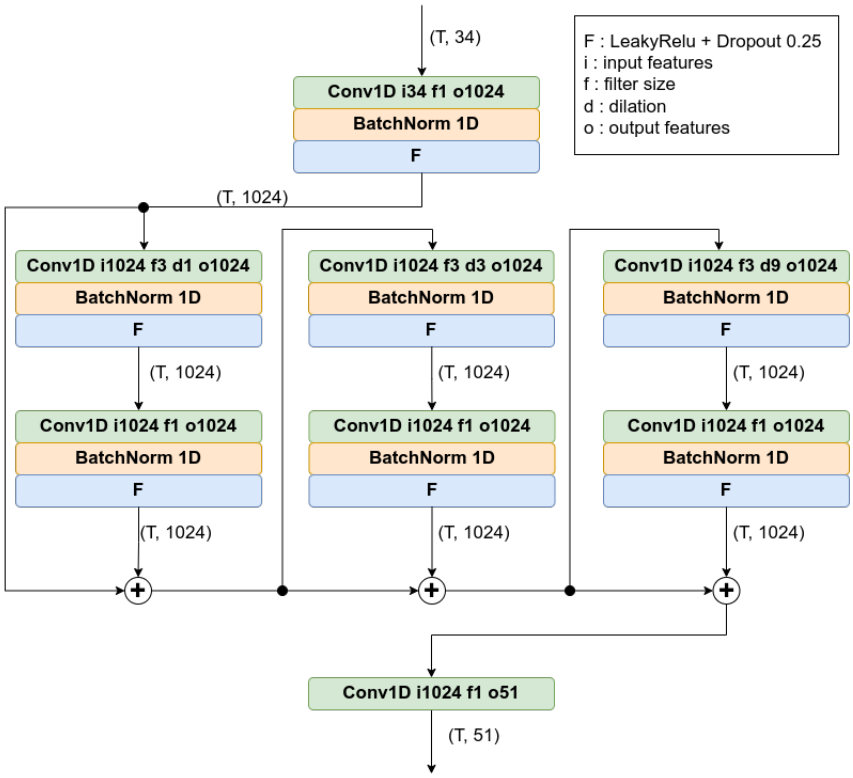

Figure 5.9 – CVM-Net architecture. Temporal Convolution Neural Networks for motion reconstruction. This is a generic approach that combines 1D convolution algorithms to transform a sequence of 2D poses into a sequence of 3D poses.

Results

To evaluate the performance of our Laplacian loss function, we have set up training-test experiments using our neural network architecture *CVM-Net*. We used the same training environment for each session with the only difference being the loss function. In these experiments, we compared two different loss functions.

— **CVM-Net** with only the joint position loss $\mathcal{L}_P$.

— **CVM-Net$_\Delta$** with a combination of the joint position loss and our *Laplacian Loss* in an overall function $\mathcal{L} = \mathcal{L}_P + \mathcal{L}_\Delta$.

The models with the two configurations above mentioned are trained under the same conditions:

— both models have the same architecture with the same parameters;

— the dataset used for the experiments is Human3.6m dataset [52];

— and, the training configuration is the same for both. It includes the learning rate of 10^{-3}, for 150 epochs. We used the optimizer **Adam** for gradients computing. The training has also been conducted on the same machine.

The evaluation results of this experiment are presented in Table 5.1. We compare the different metrics between the three trained models defined in Section 5.4, i.e. the spatial metric MPJPE and the temporal metrics MPJVE and MPJAccE.

	MPJPE (mm)	MPJVE (mm/f)	MPJAccE (mm/f^2)
CVM-Net	142.47	4.35	1.99
CVM-Net$_\Delta$	**100.62**	**3.22**	**1.33**

Table 5.1 – Comparative results of motion reconstruction with Human3.6m [52] as benchmark.

Discussion

The results of these experiments show that the *Laplacian loss* impacts both the spatial accuracy and the temporal coherence of the reconstructed motion. The improvement of the results using the metrics on velocity and acceleration proves that this loss function integrates the temporal connection between consecutive skeleton poses. With these results we conclude that our Laplacian loss function is a good choice for training deep learning solutions for HMR.

With the experiment to evaluate the efficiency of the Laplacian loss concluded, we

are now proceeding to the development of a solution to compete with state-of-the-art approaches.

5.5.2 State-of-the-Art Challenge

Here we present an alternative neural architecture to compete with state-of-the-art solutions (STAR) for human pose estimation.

We decided to develop this neural network for usage in real-time application purpose. For that reason, the model should be compact so that it can be deployed and run on systems with low computing capacity. This type of model are often called *lightweight neural network*. Generally, compact solutions have low efficiency compared to massive neural network.

We choose to benchmark our model against three different STAR models: Poseformer [140], AANet [17] and MotioNet [102]. The choice was made to cover the following conditions:

— good accuracy in predicting the positions of the skeletal joints;
— good temporal consistency of the reconstructed motion;
— preservation of the skeletal structure over time in the motion sequence;
— and a variety of approaches (sequence-to-sequence, sequence-to-pose, etc.)

MotioNet [102] is a model designed to estimate an angular representation of the motion. From a sequence of 2D poses, the approach estimates both a skeletal representation in the form of bone lengths and the changes of rotations angles (quaternions values). The estimated values are combined through a forward kinematics method to obtain the representation as a sequence of 3D poses. This method has the advantage of preserving the skeletal structure in the motion sequence, and this is also the reason why we choose to experiment with this.

The model AANet [17] uses a sequence-to-pose approach. The model is divided into two neural networks. The first neural network estimates bone directions of the central frame in the sequence while the second estimates 3D joint positions to derive the bone lengths. Both data are then combined to generate the estimated 3D pose.

Finally, the model Poseformer [140] is a transformer-based neural network and one of the best-performing models in the literature on both spatial and temporal levels. It combines two transformer modules, one spatial that extracts high dimensional feature embedding for each individual 2D pose, and one temporal that encodes dependencies across the sequence of 2D poses.

Model Architecture

The solution we designed is a variational autoencoder (VAE) neural network. We chose this type of architecture because it can compress high-dimensional data while keeping its important features. It can also capture temporal dependencies in sequential data and is suitable for reconstruction tasks [121].

The encoder takes a 2D poses sequence and outputs a latent code representation of the pose at each frame. It uses temporal convolution with a sliding window of 9 frames, meaning 9 consecutive frames are used to compute the latent code of a single frame (the central frame). For videos with a frame rate of 50Hz (Human3.6m [52]), it corresponds to 0.18s of motion to estimates the latent code of a targeted pose. The decoder is a multi-layer perceptron that uses each latent code to generate the corresponding 3D pose. Figure 5.10 shows an overview of the model architecture.

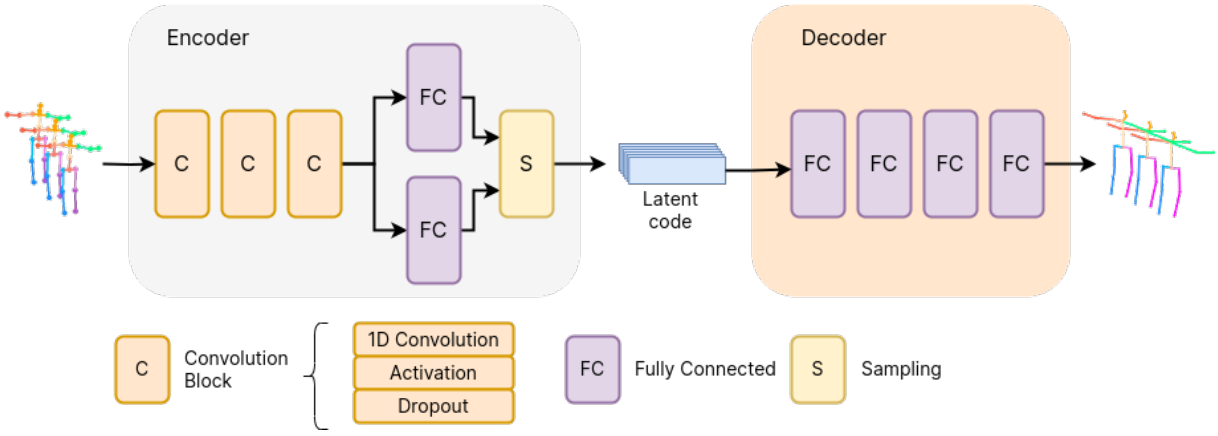

Figure 5.10 – Architecture of the motion reconstruction network based on variational autoencoder.

Implementation details

We propose a light neural network for this motion reconstruction task. Compared to the existing solutions that achieve STAR results, it has fewer parameters as shown in Table 5.2. This model does not require a GPU to run, so it is easy to deploy. We designed this solution for use on systems with low computing power.

Model	Number of parameters
Poseformer [140]	9,602,885 (9.60M)
AANet [17]	59,177,580 (59.17M)
MotioNet [102]	37,990,458 (37.99M)
Ours	218,615 (0.22M)

Table 5.2 – Comparison of model parameters.

Results

The evaluation session of our neural network is conducted with the different evaluation metrics mentioned in section 5.4. The results are then compared to the evaluation of solutions found in the literature. The following table 5.3 presents the comparative results.

	MPJPE (mm)	MPJVE (mm/f)	MPJAccE (mm/f^2)
MotioNet [102]	53.47	3.12	1.96
AANet [17]	44.63	2.64	2.21
PoseFormer [140]	30.72	1.28	0.76
Ours	88.14	3.22	0.98

Table 5.3 – Comparative results of motion reconstruction with Human3.6m as benchmark.

Discussion

The results obtained in this experiment lead to the following observations. First, our solution does not achieve better results than the selected STAR solution on the MPJPE metric related to the spatial accuracy. We may explain this with the simplicity and compactness of our model which has very few parameters compared to the others. The temporal metrics on the other hand show some interesting results. The velocity error we achieve remains in the scope of most of the STAR solutions. The acceleration error however shows a greater performance of our model than most of the STAR methods. We explain this as a result to the *Laplacian loss* function. Indeed, this loss function is based on Laplacian modeling where temporal links are set to connect each frame t with the previous and the next ones $t-1, t+1$. Note that motion acceleration at each instant t is computed between the three consecutive instants $t-1, t, t+1$. We can deduce that the representation with *Laplacian coordinates* indirectly emphasizes the acceleration of motion. Therefore

the *Laplacian loss* has greater impact on the acceleration than the velocity. As a result, we outperform many STAR solutions on the acceleration metric.

5.6 Conclusion

In this chapter, we first presented deep learning methods and how they could be applied in motion reconstruction tasks. We detailed the different steps from the formulation of the problem to the design and training of the neural network solution. We also presented the evaluation process of these solutions with the different metrics. Finally, we presented a number of experiments we have carried out to determine a high-performance solution for reconstructing motion from video. The first experiment aimed to study the impact of the *Laplacian loss* function that we proposed to guide the optimization in training of neural network model. In the second experiment, we proposed a compacted deep learning solution trained with our loss function to compete with state-of-the-art solutions.

The ablation study on the Laplacian loss has proven its efficiency in improving the spatial and temporal performance of neural network, compared to classical loss functions. The second experiment allowed us to compete with state-of-the-art approaches. Through these studies, we noted found that the acceleration in reconstructed motion was the feature most affected due to the nature of the Laplacian modeling.

The result of our state-of-the-art challenge showed that to compete with recent advances in the highly competitive research field of human pose estimation, we needed to experiment against other deep learning methods, in particular, methods with more parameters. In the remainder of our work, rather than continuing in improving this approach, we decided to use the best-performing STAR models to proceed to the second step of our methodology, that is improving the quality of motion while preserving the skeletal structure throughout sequences of 3D poses. In particular, thanks to the Laplacian loss function, we were able to confirm the effectiveness of Laplacian motion modeling. We also noticed that most STAR methods did not perform well enough on the temporal aspect. Therefore, we decided to improve the performance of existing STAR solutions, by applying a motion correction algorithm based on Laplacian modeling to the output of these models.

In the next chapter, we will propose our motion correction model to improve the spatial and temporal quality of motion estimated from video using HPE solutions.

Chapter 6

Motion Correction Systems with Deep Learning Methods and Laplacian Modeling

Contents

6.1 Introduction

Human motion data is widely used in many fields, including data-driven computer animation, motion and behavior analysis, interaction and games. This data is usually captured using optical systems with markers, mechanical or magnetic systems. However, these systems are very expensive, time-consuming to process and sometimes require a specific environment. For example, it is difficult to use a marker-based optical system in the wild.

Marker-less motion capture using cameras is one solution to these difficulties. Motion data is extracted from the outputs of RGB-D or monocular cameras. In this context, using a single monocular camera is a more challenging task. With the advent of deep learning techniques capable of learning from existing motion databases, the development of such a single-camera solution is becoming increasingly promising. The aim is to estimate 3D human poses from images or videos.

However, most current solutions do not achieve sufficient accuracy to replace traditional motion capture systems in certain application areas, such as avatar animation. In

addition, these approaches, which rely on postural proximity between estimated poses and ground truth, often overlook essential features of captured movements, such as temporal and skeletal coherence. Although there have been a few attempts to improve the temporal coherence of movement [125], or to preserve skeletal structure (even fewer approaches) [102], the proposed solutions generally focus on just one of these aspects.

Our previous reconstruction system manages to achieve good results on the temporal aspect of motion, but the results on spatial accuracy are not up to the level of the most recent solutions. In addition, all existing approaches have shortcomings with regard to the temporal aspect of motion. We therefore decided to design a motion correction system to complement 3D human pose estimation solutions so that 3D-reconstructed skeletal motion retains good spatial accuracy of 3D posture, while improving the temporal qualities of motion and preserving skeletal structure over time.

Our system is composed of:

— A fine-tuning motion module based on a Laplacian representation of motion in the form of a 3D+t graph associated with a deep learning neural network. Using this graph, motion is modeled by differential Laplacian coordinates that encapsulate the spatial and temporal characteristics of motion. The deep learning neural network refines this representation to correct the motion.

— A neural network that estimates bone lengths in a fixed way over time, enabling consistent skeletal structure during movement.

— A combined correction module. This module combines the outputs of the two previous modules, namely the corrected Laplacian differential coordinates and the corrected skeletal structure, to reconstruct motion with improved spatial and temporal quality.

In addition to the correction system, we add to our contributions, a metric to assess skeletal coherence in the reconstructed motion. It is an original skeleton-based metric to verify whether the static representation of the skeleton is preserved in the reconstructed motion over time.

Chapter outline. In this chapter, we will present these contributions and how they are combined into a motion correction system *MoCoSys*. Then we will present evaluation results of the motion correction system. Finally we will discuss about how to improve this system in future work.

6.2 Motion Correction System

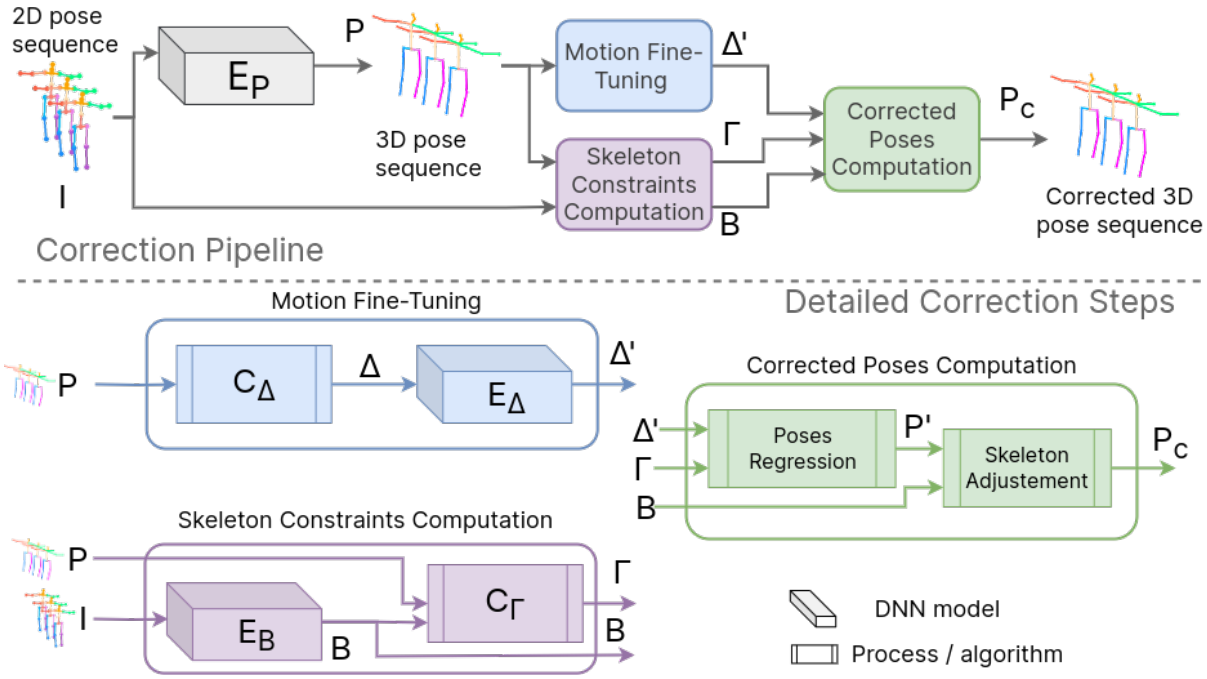

Figure 6.1 – **Motion Correction System**. Top: correction pipeline. Performed in three steps, starting with the 3D poses sequence estimated from the STAR solution E_P. Bottom: detailed correction steps. 1. *Motion Fine-Tuning*. The process C_Δ transforms the sequence of poses P into a graph, then into Laplacian differential coordinates Δ. These coordinates are corrected into Δ' using the neural network E_Δ. 2. *Skeleton Constraints Computation*. Distance constraints Γ related to the skeletal structure are computed using the neural network E_B (to compute skeleton bone lengths B) and the gamma computational algorithm C_Γ. 3. *Corrected Poses Computation*. The corrected differential coordinates Δ' and the distance constraints Γ are used to compute the corrected sequence of 3D poses.

We introduce the motion correction system with the aim of i) improving the spatial reconstruction of motion, with satisfactory temporal quality (motion fluidity) ii) minimizing the error made on the length of skeletal bones, and iii) ensuring that these bones have an almost constant length throughout the motion sequence. Figure 6.1 shows the pipeline of the approach. The top figure shows the overall architecture of our motion correction system. The figure above details the various steps of this system. Our approach builds on state-of-the-art (STAR) solutions for 3D human pose estimation from video, and then corrects the motion, both temporally and spatially. Firstly, a 3D+t representation of the motion [63], coupled with a deep learning approach, is used to estimate the differential coordinates of the Laplacian (*Motion Fine-Tuning* module). The second step (*Skeleton*

Constraints Computation module) adds distance constraints related to the skeleton structure. Finally, the third step (*Corrected Poses Computation* module) gives result to the corrected sequence of 3D poses by combining the outputs of the two previous modules.

6.2.1 Step I: Motion Fine-Tuning

In this stage, our aim is to fine-tune the motion represented as a Laplacian graph 3D+t in order to improve the spatio-temporal features. First we convert the estimated sequence of 3D poses P into Laplacian differential coordinates with the *Delta Conversion Unit* C_Δ. Then, the neural network E_Δ estimates corrected differential coordinates Δ' which are then used in the final Corrected Poses Computation module.

Delta Conversion Unit C_Δ

This unit uses the discrete Laplacian operator corresponding to the 3D+t motion graph to compute the differential coordinates Δ of the skeleton joints as defined above. For a sequence of T poses, each skeletal pose being composed of J joints, we have a total of $N = T \cdot J$ joints in the graph $G = (V, E = E_S \cup E_T)$. The adjacency matrix A of size $(N \times N)$ associated to the graph is defined by:

$$A_{ij} = \begin{cases} 1 & if (i,j) \in (E_S \cup E_T) \\ 0 & otherwise \end{cases} \tag{6.1}$$

Using the degree matrix D and the identity matrix I, we can compute a simple version of L as:

$$L = I - D^{-1}A \tag{6.2}$$

Finally, the matrix Δ of dimension $(N \times 3)$ of differential coordinates can be computed as:

$$\Delta = LP \tag{6.3}$$

with P the stacked matrix of dimension $(N \times 3)$ of the 3D coordinates of the graph joints.

Neural Network E_Δ

This model consists of layers of neural networks that learn to correct motion in differential coordinate space. We have chosen a network architecture based on graph convolution [57], using the 3D+t graph structure of motion. This can be summarized by the function:

$$\Delta' = E_\Delta(\Delta) \tag{6.4}$$

Figure 6.2 illustrates the architecture of E_Δ.

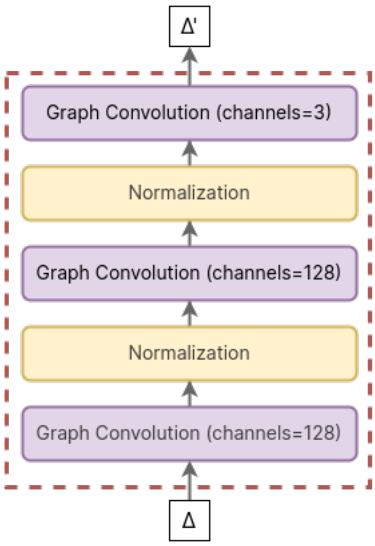

Figure 6.2 – **Learning Block** E_Δ. It consists of a sequentially connected graph convolution neural network.

As it operates on the differential coordinates Δ, the learning block E_Δ is trained to minimize the associated loss. We call this loss the *Laplacian loss* [114]. It is defined as the average distance between the corrected and actual differential coordinates according to:

$$\mathcal{L}_\Delta = \frac{1}{N}\sum_{1}^{N} \|\Delta^{gt} - \Delta'\| \tag{6.5}$$

where N is the total number of joints, Δ^{gt} is the matrix of the ground truth differential coordinates, and Δ' the matrix of corrected differential coordinates estimated by E_Δ.

6.2.2 Step II: Skeleton Constraints Computation

The Laplacian differential coordinates Δ' obtained in the first stage do not characterize the skeletal structure of the 3D+t graph of motion in Euclidean space. Rather, they are a local representation based on the distance of each joint from its direct neighbors in Laplacian space. Therefore, to obtain the corrected graph in the form of 3D skeletal poses from Δ', we need constraints between joints related to human skeletal structure expressed in Euclidean space. We start by estimating the lengths of the skeletal bones (fixed distances between skeletal joints) using the neural network E_B. We then compute these skeletal constraints Γ through C_Γ.

Neural Network E_B

The neural network E_B is in charge of estimating fixed lengths B for the skeleton bones. These lengths are then used to define vector constraints between skeletal joints to ensure consistency of skeletal structure throughout motion. Figure 6.3 shows the architecture of the neural network E_B. E_B estimates the skeleton bone lengths B, taking into account

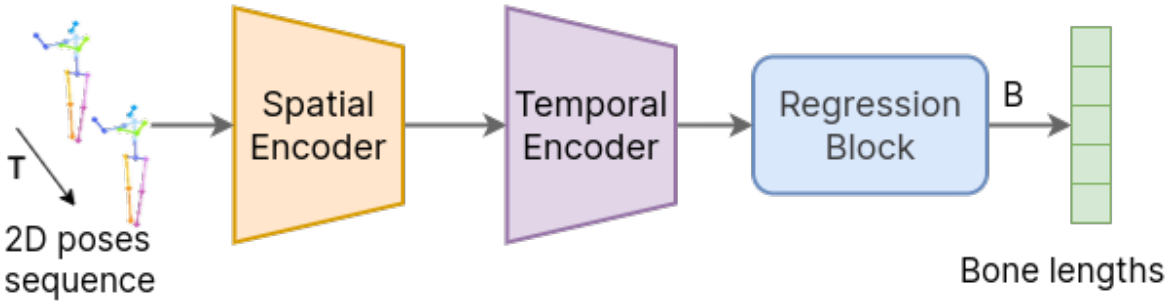

Figure 6.3 – E_B **Neural network architecture**. It consists of convolution layers, grouped into 3 blocks. The first block is a spatial encoder, the second a temporal encoder, and the third one is a regression block that collapses the temporal axis. The output of this NN is the static representation of the skeleton (bone lengths B).

the fact that symmetrical bones should have the same length. Table 6.1 presents in detail the architecture of the model.

Gamma Computational Unit C_Γ

We represent by Γ the differential vectors integrating the distance constraints between the skeletal joints. The matrix *Gamma* of all the skeletal bone vectors is computed by normalizing each vector to be corrected so that they all have the desired length (lengths

Name	Layers	k	s	in/out
Spatial Encoder	(Conv + ReLU + Drop)	1	1	34/32
	(Conv + ReLU + Drop)	1	1	32/32
Temporal Encoder	(Conv + ReLU + Drop)	3	1	32/32
	(Conv + ReLU + Drop)	3	1	32/32
Regression Block	(Batch Normalization + Average Pooling + Conv)	1	1	32/10

Table 6.1 – Detailed architecture of E_B model. k represents the kernel size and s the strides of the convolution. The input and output channels are also given. A skeleton of 17 joints is composed of 16 bones. If we consider symmetrical bones, we retain only 10 values.

obtained from the bone lengths B). The resulting vectors Γ are computed through the Algorithm 1.

Algorithm 1 Algorithm to compute Γ

Data: P: Set of joint position coordinates from the motion
B: Dictionary of lengths for each skeleton bone
Results: Γ: Set of bone vectors with correct lengths
$\Gamma \leftarrow$ Set of all bone vectors computed from P ;

for $\mathbf{b} \in \Gamma$ **do**
    $\mathbf{b} \leftarrow B[name(\mathbf{b})].\frac{\mathbf{b}}{||\mathbf{b}||}$ ▷ Corrects each bone vector to have the desired length
end for

We note $\Gamma = C_\Gamma(P, B)$.

6.2.3 Step III: Corrected Poses Computation

This final step of the correction system calculates the corrected motion in the form of a sequence of 3D skeleton poses, by solving a constrained linear system using the results of the previous steps.

Poses Regression

This module computes the new positional coordinates P' of the joints from the corrected differential coordinates Δ' computed in stage I and the distance constraints Γ obtained in stage II. We obtain P' by solving the linear system represented by the matrix equation $LP' = \Delta'$. However, solving this linear system leads to an infinite number of

solutions, some of which do not preserve the skeletal structure. As a solution to this issue, we choose to over constrain this system. Two types of constraints are therefore considered. Firstly, we introduce root position constraints to reduce the number of solutions due to the translation-invariant property of the discrete Laplacian operator. We thus compute the solution with a fixed root position (pelvis) for each skeleton in the sequence. Secondly, to preserve skeletal structure, we define vector constraints between joints.

Root Position Constraints These constraints will ensure that the root of all the skeletons are set to $(0, 0, 0)$. Let $V_R \subset V$ the set of joints that represents the pelvis of the skeletons within the sequence of poses and $V_t \subset V$ the set of joints of skeleton S_t (see 4.3.2). The constraints are defined as: $p'_i = (0, 0, 0)$ if $v_i \in V_R$ and $v_i \in V_t, 1 \leq t \leq T$, where p'_i is the desired Cartesian coordinates of vertex v_i. To integrate these constraints into the above equation system, we add the constraint equation system $UP' = R$, in which U is a sparse matrix that extracts the coordinates vectors of the root from P'. It is a matrix $(T \times N)$ defined by:

$$U_{ti} = \begin{cases} 1 & \text{if } v_i \in V_R \text{ and } v_i \in V_t \\ 0 & \text{otherwise} \end{cases} \tag{6.6}$$

The R matrix represents the values assigned to the root positions in P'. It is a column matrix of vectors $(0, 0, 0)$.

Skeleton vector constraints To preserve the structure of each skeleton, we add vector constraints on the bones composing the skeleton. Each constraint is defined as a vector between two joints of the same skeleton, corresponding to the edges in E_S. For each $(v_i, v_j) \in E_S$, we note $\gamma_{ij} = p_i - p_j$, the directional vector between joints v_i and v_j. We have a total of $card(E_S)$ vectors. We can stack them into a matrix Γ of size $card(E_S) \times 3$. Let D an operator that computes the vectors Γ from the coordinates P. D is a $(card(E_S) \times N)$ matrix defined as:

$$D_{ek} = \begin{cases} 1 & \text{if } k = i \\ -1 & \text{if } k = j \\ 0 & otherwise \end{cases} \tag{6.7}$$

with $e = (v_i, v_j) \in E_S$.

Using the vectors obtained from the gamma computational unit C_Γ, we have the

desired values for skeleton constraints. We can then define the constraints $DP' = \Gamma$.

Joint Position Computation Adding the above-mentioned constraints, we have the final equation system to compute the joint positions:

$$\begin{pmatrix} L \\ U \\ D \end{pmatrix} P' = \begin{pmatrix} \Delta' \\ R \\ \Gamma \end{pmatrix} \tag{6.8}$$

This system is an over-determined linear system and therefore requires a least-squares approximation method for its resolution. To solve this system, we opted for Cholesky factorization. The resulting solution vector is the sequence of corrected poses.

Note that the Cholesky decomposition matrix can be time-consuming if the sequence is too large, or if we need to perform it for several sequences of different sizes. Therefore, to optimize performance, the process is performed in sub-sequences using a fixed-size sliding window.

Skeleton Adjustment

Cholesky factorization produces an approximate result that adjusts both the coordinates of the Laplacian Δ' and the constraints of the skeleton vector Γ. As a result, the constraints of the skeleton structure are applied as soft constraints. To solve this issue, we add an additional process that strongly enforces these constraints. From the previously obtained joint positions P' and skeleton bone lengths B, we compute new distance constraints $\Gamma_c = C_\Gamma(P', B)$.

The final P_c coordinates of the joint positions are obtained by solving the following system of linear equations:

$$\begin{pmatrix} U \\ D \end{pmatrix} P_c = \begin{pmatrix} R \\ \Gamma_c \end{pmatrix} \tag{6.9}$$

This linear system has the same number of equations and unknowns, and therefore a unique solution.

6.2.4 Motion Correction System

The algorithm 2 summarizes the complete process of our approach.

Algorithm 2 Motion Correction Algorithm

Data: I: Sequence of 2D joint positions

Results: P_c: Sequence of corrected 3D poses

$P \leftarrow E_P(I)$

▷ Step I: Motion Fine-Tuning

$\Delta \leftarrow C_\Delta(P)$

$\Delta' \leftarrow E_\Delta(\Delta)$

▷ Step II: Gamma Computation

$B \leftarrow E_B(I)$

$\Gamma \leftarrow C_\Gamma(P, B)$

▷ Step III: Corrected Poses Computation

▷ i): Delta-Gamma Fitting

▷ Compute P' from Cholesky Resolution

$$\begin{pmatrix} L \\ U \\ D \end{pmatrix} P' = \begin{pmatrix} \Delta' \\ R \\ \Gamma \end{pmatrix}$$ ▷ equation 6.8

▷ ii): Skeleton Adjustment

$\Gamma_c \leftarrow C_\Gamma(P', B)$

▷ Compute P_c from Linear Resolution

$$\begin{pmatrix} U \\ D \end{pmatrix} P_c = \begin{pmatrix} R \\ \Gamma_c \end{pmatrix}$$ ▷ equation 6.9

6.3 Experiments and Results

6.3.1 Additional Evaluation Metrics

In the process of correcting motion, we pointed out a requirement that needs to be fulfilled, i.e., the consistence of the skeleton. To ensure that the skeleton is unchanged through the motion sequence, we define a new metric, called the *Skeleton Variation Error*. This metric, called SVE, is computed as the mean of standard deviations of bone lengths over the motion sequence. Let us consider a motion composed of T frames. The following formula gives the standard deviation of a bone b:

$$\sigma_b = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(d_{b,t} - \mu_b)^2} \tag{6.10}$$

with $d_{b,t}$ representing the length of the bone b of the skeleton at time t and μ_b the mean length of bone b.

We then compute skeleton consistency metric as follows:

$$SVE = \frac{1}{card(\mathcal{B})}\sum_{b\in\mathcal{B}}\sigma_b \tag{6.11}$$

with $\mathcal{B}$ the set of bones of the skeleton. The closer this metric is to 0, the more consistent the skeleton is within the reconstructed motion.

6.3.2 Quantitative Evaluation

Evaluation Methodology

Dataset We evaluate our approach with the well known dataset Human3.6M [52]. It is a dataset of 3D human poses captured indoor with a marker-based motion capture system. The data consists of 15 different daily actions executed by 11 professional actors and recorded by 4 cameras. It contains 3.6 million captured video frames annotated with 3D positions and rotations (ground truth) at a frequency of 50 frames per second. In line with the evaluation process for all related work, the training set contains data for 5 subjects (S1, S5, S6, S7 and S8) and the test set for 2 subjects (S9 and S11). All actions are included in the sets.

Experiments Details The experiments take place in two phases. In the first phase, the neural networks for each correction step are trained as follows:

1. For the motion fine-tuning step, the E_Δ neural network is trained to learn better Laplacian representation of the 3D pose sequences estimated using the STAR estimator. To avoid the experiment being biased by the efficiency of the estimator, we choose one that does not perform best on either temporal or skeletal coherence. In this experiment, we used Chen et al. [17] 3D pose estimator (AANet) to train E_Δ. Algorithm 3 presents a pseudo code of the training process.

Algorithm 3 E_Δ training process

Data: D: Training dataset
for n epochs **do**
  for $I, P_{gt} \in D$ **do**
    $P \leftarrow E_P(I)$
    $\Delta \leftarrow C_\Delta(P)$
    $\Delta_{gt} \leftarrow C_\Delta(P_{gt})$
    $\Delta' \leftarrow E_\Delta(\Delta)$
    $l_\Delta \leftarrow \mathcal{L}_\Delta(\Delta', \Delta_{gt})$
    Compute gradients from l_Δ
    Update weights of E_Δ
  end for
end for

2. For the skeleton constraints computation step, the neural network E_B, is trained to estimate skeleton bone lengths from the 2D joint locations in the benchmark. The skeleton bone lengths of the corresponding 3D poses are used as ground truth for the supervised learning.

Finally, we conduct the evaluation of the whole correction system with the previously trained neural networks. The validation process can then take place in two stages:

— a simple validation using the same STAR estimator as in the training session;

— a cross-validation where we replace the estimator with another STAR solution.

Each evaluation is performed on the whole test set of the Human3.6M dataset [52], using as input the 2D ground truth. We compare the motion reconstructed by the STAR estimator and the motion reconstructed after the correction steps.

We use three different STAR estimators to evaluate our correction system.

Estimator AANet AANet [17] is the solution used to train the E_Δ neural network of the motion correction stage.

Metrics	AANet [17]	AANet [17] + Correction	Δ	Δ%
MPJPE (mm)	**44.63**	44.88	↑0.25	↑0.87
MPJVE (mm/f)	2.64	**2.27**	↓0.37	↓ 14.01
MPJAccE (mm/f^2)	2.21	**1.00**	↓ 1.21	↓ 54.75
MBLE (mm)	7.70	**3.76**	↓ 3.94	↓ 51.17
SVE (mm)	1.79	**0**	↓ 1.79	↓ 100

Table 6.2 – Results of the motion correction system with AANet [17].

Estimator MotioNet MotioNet [102] is a solution that ensures the skeletal consistency of the reconstructed motion.

Metrics	MotioNet [102]	MotioNet [102] + Correction	Δ	Δ%
MPJPE (mm)	53.47	**52.85**	↓ 0.62	↑ 1.15
MPJVE (mm/f)	3.12	**2.73**	↓ 0.39	↓ 12.50
MPJAccE (mm/f^2)	1.96	**1.22**	↓ 0.74	↓ 37.75
MBLE (mm)	**3.08**	3.76	↑ 0.68	↑ 22.07
SVE (mm)	0	0	0	0

Table 6.3 – Results of the motion correction system with MotioNet [102].

Estimator PoseFormer PoseFormer [140] is a solution of the literature that achieves excellent results in terms of both spatial accuracy and temporal consistency.

Metrics	PoseFormer [140]	PoseFormer [140] + Correction	Δ	Δ%
MPJPE (mm)	30.72	**29.71**	↓ 1.01	↓ 5.24
MPJVE (mm/f)	**1.28**	1.34	↑0.06	↑ 0.78
MPJAccE (mm/f^2)	0.76	0.76	0	0
MBLE (mm)	8.20	**3.76**	↓ 4.44	↓ 54.14
SVE (mm)	4.92	**0**	↓ 4.92	↓ 100

Table 6.4 – Results of the motion correction system with PoseFormer [140].

6.3.3 Discussion

Spatial accuracy Our correction system does not degrade the error on the position of the joints and sometimes even improves it a little. The MPJPE results show that our system can be adapted to any STAR method chosen as the basis for 3D pose estimation.

Skeleton structure The lengths of the bones that characterize the skeletal structure of the human should not change through the motion. In STAR methods, skeletal structure in motion is generally neglected. The results show that our correction system guarantees that this aspect is always verified (the SVE is always 0). Our correction system also reduces the average error in bone length to 3.76 mm.

Temporal coherence The temporal aspects are also greatly improved compared to the chosen state-of-the-art solutions. This is an important factor to take into account, depending on the subsequent use of the reconstructed motion (whether motion analysis or synthesis). The closer the acceleration and velocity errors (MPJAccE and MPJVE) are to 0, the closer we come to the fluidity of the original motion.

Visual results In Figure 6.4, we can observe in more details some comparative results. The acceleration and velocity curves show that the correction system improves the temporal quality of motion. The acceleration curves display the most significant results with the error curve closest to ground truth. The velocity curve is smoother, with variations similar to the ground truth curve. These observations show that our corrected motion achieves a smoothness very close to that of the ground truth. Furthermore, by observing the curves related to variations in bone length, we can affirm that skeletal consistency has been achieved and that bone lengths are closer to the ground truth.

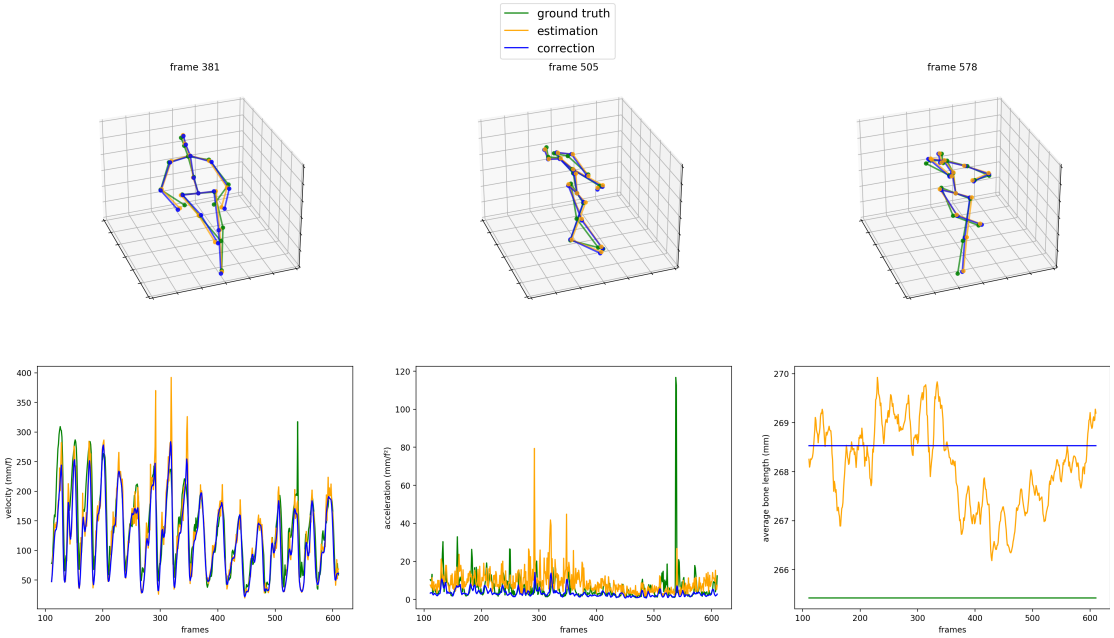

Figure 6.4 – Example of comparative results, action: walking. In green we have the ground truth motion, in orange the motion from the STAR model AANet [17] and in blue the motion after correction. Top: we have the skeleton poses from the estimation, the correction and the ground truth at different frames. Bottom: from left to right we have the velocity curve, the acceleration curve and the average bone length curve. The different curves show that using our correction system brings better quality results than the estimation.

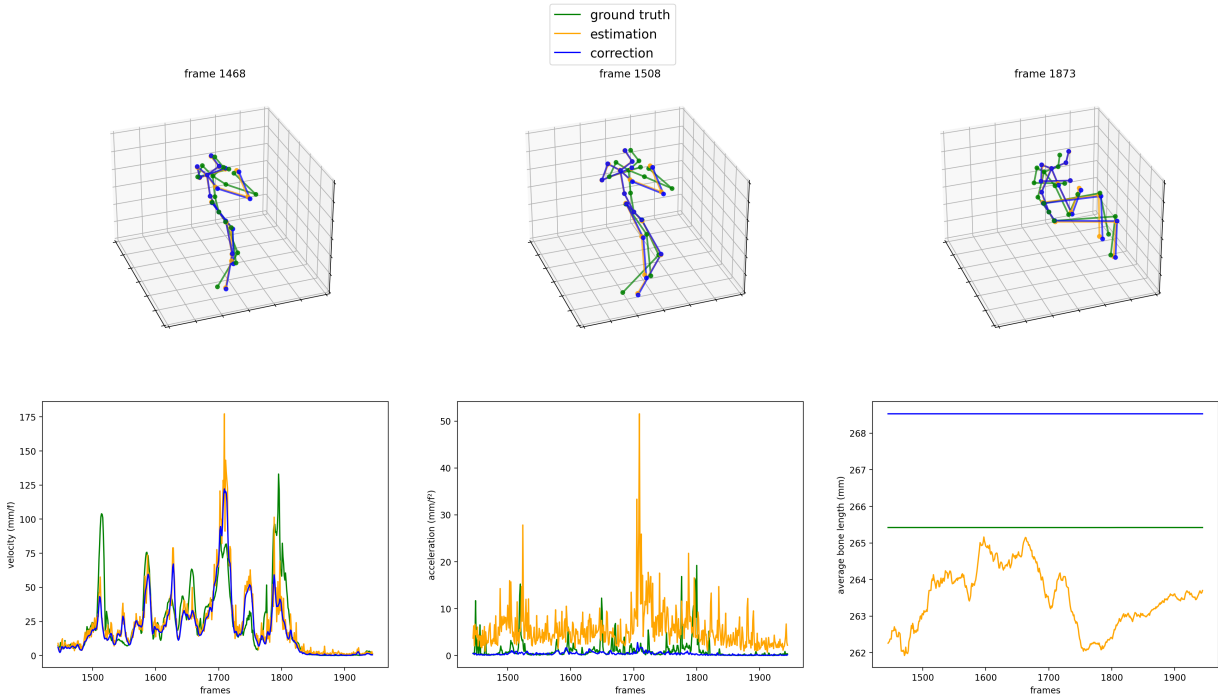

Figure 6.5 – Additional comparative results, action: phoning.

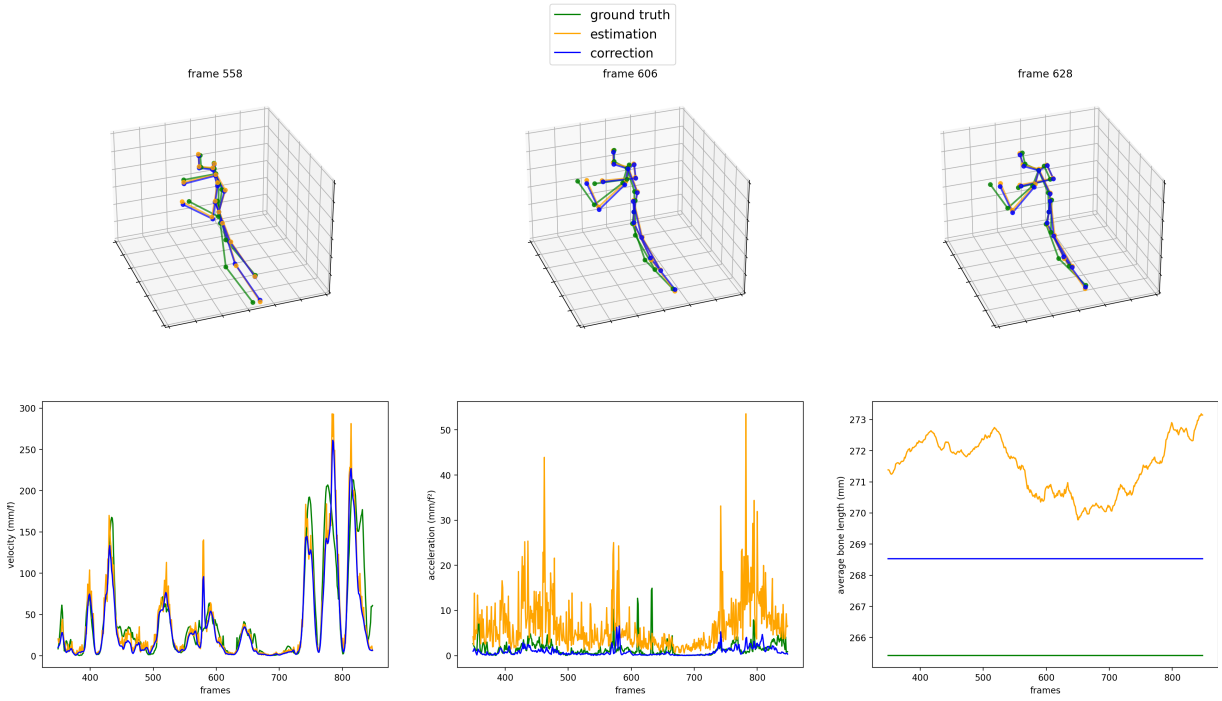

Figure 6.6 – Additional comparative results, action: discussion.

6.3.4 Ablation Study on MoCoSys

In this study we explore the impact of the correction process. We specifically study the impact of the *skeleton adjustment* process in the *corrected poses computation* step. We use as baseline a STAR method, and observe the results obtained after the correction process. Table 6.5 shows the results of the study with three different baselines. These results show that the correction module achieves the objectives assigned to it.

	MPJPE (mm)	MPJVE (mm/s)	MPJAccE (mm/s^2)	MBLE (mm)	SVE (mm)
Baseline	**44.63**	2.64	2.21	7.70	1.79
Baseline + Correction(PR)	44.78	2.28	1.02	7.99	2.56
Baseline + Correction(PR + SA)	44.88	**2.27**	**1.00**	**3.76**	**0**

(a) Baseline = AANet [17]

	MPJPE (mm)	MPJVE (mm/s)	MPJAccE (mm/s^2)	MBLE (mm)	SVE (mm)
Baseline	30.72	**1.28**	**0.76**	8.20	4.92
Baseline + Correction(PR)	31.58	1.33	**0.76**	8.56	5.06
Baseline + Correction(PR + SA)	**29.71**	1.34	**0.76**	**3.76**	**0**

(b) Baseline = PoseFormer [140]

	MPJPE (mm)	MPJVE (mm/s)	MPJAccE (mm/s^2)	MBLE (mm)	SVE (mm)
Baseline	53.47	3.12	1.96	**3.08**	**0**
Baseline + Correction(PR)	**52.61**	**2.72**	**1.22**	4.57	2.49
Baseline + Correction(PR + SA)	52.85	2.73	**1.22**	3.76	**0**

(c) Baseline = MotioNet [102]

Table 6.5 – Ablation study. We explore the impact of the skeleton reinforcement step. (a), (b) and (c) use different baselines. PR=Poses Regression, SA=Skeleton Adjustment. The correction system produces good results whatever the baseline.

The motion correction system significantly improves temporal metrics, as shown by the reduction in MPJVE and MPJAccE errors (respectively for velocity and acceleration), while showing little or no degradation in MPJPE.
With the **skeleton adjustment** operation, the correction reduces the error on bone lengths (MBLE) to 3.76 mm, which is better than almost all existing solutions. The

skeleton variation error SVE is also reduced to approximately 0, showing that the skeletal structure is preserved.
We can conclude that our motion correction system significantly improves 3D motion reconstruction. Indeed, the average error on bone length is reduced, skeletal structure is preserved over time thanks to the skeletal correction module, and errors on velocity and acceleration are reduced thanks to our 3D+t fine-tuning module.

6.4 Conclusion

In this chapter we presented a system for motion correction based on deep learning applied on a 3D+t Laplacian modeling of motion. The system is used in conjunction with an existing method for estimating 3D pose from video. It improves 3D pose estimation and motion quality by preserving skeletal structure and enhancing temporal smoothness. The system consists of two deep neural networks. The first neural network estimates the static features of the skeleton, namely the bone lengths. The second neural network works on the Laplacian representation of the estimated movement to correct the local deformation in time and space of each joint. Finally the system uses both the estimated skeleton model and the corrected Laplacian representation to reconstruct a smoother, less noisy motion. Thanks to our approach, the captured motion meet the requirements of a wider range of data-driven applications, particularly those involving motion synthesis.

There are some limitations to the method and we have some ideas for improvement. Our current system is based on an intermediate step of estimating 2D joint positions before estimating skeletal bone lengths. As a result, estimation errors accumulate. In future work, we aim first to improve and generalize the estimation of skeletal bone lengths by using a neural network that directly estimates skeletal structure from video. Secondly, in the motion correction process, the unique solution of the linear system is obtained thanks to the Γ vector constraints on the skeletal structure, calculated from the estimated 3D poses. We intend to find new ways of computing the graph representation from differential coordinates without depending on estimated 3D poses, probably by learning Γ through deep learning, directly from video or 2D joint positions, or by defining prior information on skeletal structure.

Part III

Motion Dataset and Application to Motor Disability

Chapter 7

Handi-Motion: A Database of Motions from People in Motor Disability State

Contents

7.1 Introduction

In recent years, the perception of disability and the attention given to people in deficiency situation has considerably evolved, opening the way to new disciplines. Numerous researchers and engineers, in collaboration with healthcare professionals, are now involved in the development of technologies designed to facilitate the autonomy and improve the living conditions of people with disabilities. In this work, we are specifically interested in applications related to people with motor deficiencies that their movements and require the use of a mobile wheelchair [54, 27].

Many applications related to human motion are based on methods and tools that rely on the use of data. However, despite current technologies and methods, collecting motion data from wheelchair users remains difficult to realize. As a result, in clinical practice,

human motion analysis is mainly limited to quantified gait analysis (QGA) to meet the needs of different audiences [25, 31].

The best solution to collect motion data is to use motion capture systems (MoCap) for highly precise capture. But, these systems are sometimes restrictive and require recording conditions that are difficult for wheelchair users to satisfy. For example, with marker-based capture systems, it is difficult to obtain a reference pose, usually the standing T-pose, which enables post-processing operations required to obtain clean and consistent data.

Video-based motion reconstruction techniques are relatively imprecise and present a number of problems, including occlusion problems due to the small number of cameras used. In addition, most of these techniques require the use of existing motion data to train AI models before using them. However, such data is not currently available.

Despite the difficulties encountered when using traditional MoCap systems, we sought a compromise that would make the use of these systems possible. To this end, we chose to capture the movements of participants with mild disabilities who were able to stand up for a few seconds. With such participants, it becomes possible to perform calibration, which facilitates the post-processing of the captured data.

Our aim is to propose a corpus of movement data to enable the development of applications related to motor disability. More specifically, we aim to build a *MoCap* corpus that will then be extended by synthesis to train artificial intelligence models. The ultimate goal is to automatically reconstruct the movements of wheelchair users in a variety of environments, particularly in the context of home care. This makes it possible to implement automatic supervision applications, requiring action recognition and anomaly detection, such as fall or inactivity detection. In the context of motor rehabilitation, semi-automatic supervision can also be envisaged, in particular for analyzing and interpreting data and facilitating therapeutic diagnosis, in order to assess the motor capacities of people in handicap situations. In the same context, we could also mention applications for tele-education, assessment of independence levels or analysis of functional gestures. Our research into the analysis and synthesis of movements aims to meet these different objectives.

This chapter presents *Handi-Motion*, a *Motion Capture* corpus defined for motion from people with physical disabilities. In section 7.2, we describe the motivations and the content of this corpus. Section 7.3 presents the protocol we followed to acquire the data while section 7.4 explains the format of the captured data. Finally, section 7.6 presents an experiment we realize on this database in line with the original purpose of this thesis.

7.2 Corpus Definition

Large *Motion Capture* databases cover a large variety of motion categories. But even those databases can cover neither all motion categories nor all possible actions related to one category. This is due to the time and resources needed to acquire motion data with traditional *MoCap* systems. Therefore, for each database, a data corpus is defined to detail a range of motion to cover. For example, the **Fit3D** database [36] specifically targets motion data related to physical and fitness exercises.

The corpus definition specifies not only the objectives and needs that motivate the construction of a database but also details the content of the database.

Our corpus concerns movements from motor disability situations. We especially target disabilities that require the use of a wheelchair. We based our selection in a situation of home care, with reference to internationally recognized classifications and models in this field, notably those of the ICF (International Classification of Functioning, Disability and Health) [109, 108].

7.2.1 Motivations

There are 4 reasons for designing this data corpus.

Our primary motivation is to have data available to train artificial intelligence (AI) models for motion reconstruction, as well as action detection and recognition. These models will be used to support home monitoring systems in connected apartments for people with disabilities. The information gathered by these systems will be used to analyze daily activities in order to improve living conditions in these apartments.

Secondly, some actions are performed differently according to the type of disability. With different disabilities, there is a wide variety of gestures that can be obtained for a single action. Through this corpus, we aim to identify and analyze the differences that may arise in the execution of different actions. This will help to study the impact these differences have on AI models for action recognition and to test their robustness.

Thirdly, traditional motion capture methods are difficult to use for wheelchair users. These methods often impose constraints that are difficult to meet in situations of motor disability, such as the need to fit special suits or tools, the need to stand in a T-posture (standing, with arms straight, forming the letter T) for calibrating the systems, and many others. For optical capture with markers, there are also problems of occlusion due to the presence of the chair. All these constraints can make data post-processing long

and tedious, depending on the technology used and the type of action performed. Finally, using these systems requires technical skills that are difficult to acquire and test in a clinical setting. For all these reasons, the creation of the Handi-Motion corpus enables experimentation with new AI methods for motion reconstruction from RGB video (and by extension RGB-D), which are a priori less restrictive than MoCap methods and easier to exploit. In particular, these AI models will be used to set up real-time motion tracking and recording systems, facilitating clinical practice where professionals work in limited time. Indeed, one of our objectives is to be able to generate videos of movement in situations of disability from the MoCap data in this corpus. These will be used to train AI models for motion reconstruction.

Finally, this data corpus will, like any existing one, be a source for avatar synthesis and animation, motion analysis and many other motion data-driven applications.

7.2.2 Content of the Corpus

The prerequisite before capturing motion data is to define what categories of motion to capture based on one's need. For that, we worked in collaboration with professionals (home occupational therapists, engineers specializing in technological assistance and smart home) using a methodology comprising the following stages:

1. Review of ICF (International Classification of Functioning) items ;
2. Focus group and interviews with experts to select items/scenarios related to movement and home;
3. Prioritization of items/scenarios based on the analysis capabilities of generic video capture systems and the issues identified in the scientific literature in the field.

We first identified a list of actions related to the use of a mobile wheelchair. As our primary aim is building AI models for home monitoring, we selected a number of actions that govern daily life, grouped into three classes.

— **Mobility**. This category covers all wheelchair-related actions for moving around, including the following : *move forward, backward, turn (left or right), turn around, stop* ;

— **Daily gestures**. In this category are listed daily actions executed by mobile wheelchair user among which: *grasping/dropping/throwing an object, stretching, opening specific objects (boxes), opening a door, eating, drinking, putting on shoes, putting on clothes, doing push-ups* ;

— **Transfer**. These are specific actions that consist of leaving the wheelchair for either sitting on a chair or lying down in bed and vice versa. We list: *bed-wheelchair transfer, wheelchair-chair transfer.*

Combining these actions, we wrote a total of 12 scenarios of motion that will serve as a guiding line during the data acquisition sessions. They are the following:

1. **Drink a glass of water** : (1) open a high cupboard on the left-hand side; (2) pick up a glass and put it on the table in front of you; (3) help yourself to a drink from a carafe on the right-hand side of the table; (4) bring the glass to your mouth and drink.
2. **Changing posture in the wheelchair**: (1) do a wheelchair push-up (5 seconds); do some stretches.
3. **Cutting a food item** : (1) pick up a food item (ex: a banana, on the right-hand side of the table); (2) put the food item in the cutting board in front of you; (3) pick up a knife (right-hand side of the table) and cut the food item; (4) pick up the cut ends of food with your hands and put them on a plate (left-hand side of the table).
4. **Help yourself to food** : (1) pick up the cutlery and place them on either side of the plate in front of you; (2) take the salad bowl on the right of the plate in front of you; (3) serve some salad on the plate; (3) pick the cutlery and cut the salad; (4) prick some salad with the fork and bring the fork to your mouth.
5. **Brushing your teeth**: (1) move to the front of the sink; (2) grab the toothbrush and toothpaste (placed at the back of the sink); (3) apply some toothpaste to toothbrush; (4) brush your teeth ; (5) rinse your mouth with a glass of water (placed to the left of the sink).
6. **Pick up a pencil** : (1) remove the brakes of the wheelchair; (2) move a little forward; (3) turn to the right; (4) stop and put on the brakes; pick up the pencil on the ground.
7. **Wear a jacket**: (1) move forward to the front of the closet; (2) open the closet while moving a little backward; (3) pick a jacket in the closet; (4) put on jacket and zip up; (5) wait; (6) zip down and take off the jacket.
8. **Go to bed** : (1) move from the wheelchair to the bed; (2) lie down on your back; (3) turn to lie down on your right side; (4) wait; (5) turn back to lie down on your back; (6) wait; (7) position yourself correctly in the bed using the support arm.

9. **Having breakfast** : (1) move toward the work plan in the kitchen; (2) take two slices of bread and put it in the toaster; (3) activate the toaster; (4) wait; (5) remove the bread and put it on your knees; (6) turn around and move back to the dining table; (7) put the bread on the table, (8) pick a knife and spread the butter on the bread; (9) open jam pot and spread jam on bread.
10. **Getting up in the morning** : (1) transfer from bed to wheelchair; (2) remove the brakes and move forward; (3) turn off the light by flipping the switch; (4) open the door in front; (5) move backward while closing the door; (6) turn around and move forward.
11. **Fall simulation**: Perform various fall scenes. *(Performed only by APAS -Adapted Physical Activity or Sport- students among the actors)*
12. **Free scene**: Participants are free to performed actions they want. The purpose of this scenario is to record natural movements without constraints.

7.3 Data Acquisition

Once the data corpus had been defined, it was necessary to find a rigorous protocol for carrying out the capture. This protocol took into account the technical parameters, which included all aspects relating to the equipment to be used. We also had to consider solutions for capturing the movements of a sufficient number of participants, given the constraints associated with MoCap.

7.3.1 Technical Parameters

Data acquisition was carried out using traditional capture methods. We opted for marker-based optical capture and chose the *Optitrack-Motive* system. Infrared cameras are used to locate the various markers. The marker information is then used to reconstruct the movement. The technical parameters therefore take into account the capture environment as well as the choice of a set of markers to be positioned on the body.

Motion Capture Room

The movements in the corpus require sufficient space to facilitate wheelchair movements. The capture sessions took place in a gymnasium equipped with the *Optitrack* system. The room was about 100 m^2 in size and housed 18 MoCap cameras. Figure 7.1

shows the dimensions of the room and the positioning of the MoCap cameras. The cameras capture at a frequency of 120 fps with an accuracy down to 0.45mm after calibration.

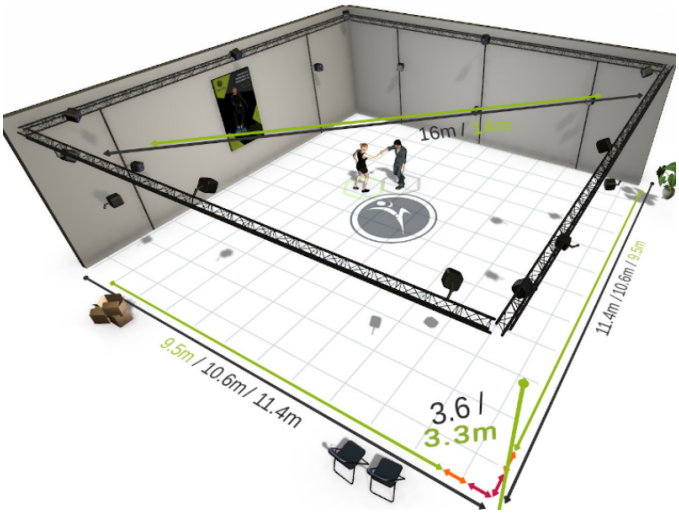

Figure 7.1 – MoCap Room.

Marker Set

For this motion capture, we used the pair *Optitrack-Motive* which is a marker-based motion capture system using passive markers attached to the dedicated suit worn by the actor. In order to obtain the best recordings, it is imperative to define a *marker set*, meaning the number of markers to use and their respective positions on the body. We used for this MoCap database a set of 49 markers arranged on the body as shown in Figure 7.2.

7.3.2 Participants

The ideal protocol for acquiring these data would be to capture only those people who are truly motor-impaired. However, the settings required by the MoCap technology make it difficult for them to be recorded. In fact, the system needs a calibration step where the actor is asked to stand in T-Pose for a certain period, at best 5 seconds if the pose is correctly executed. This step is not easily achievable for everyone with motor difficulties. The capture was therefore carried out on 8 participants, 3 of whom were in real situations of motor impairment, the other 5 simulating the scenarios. There are 2

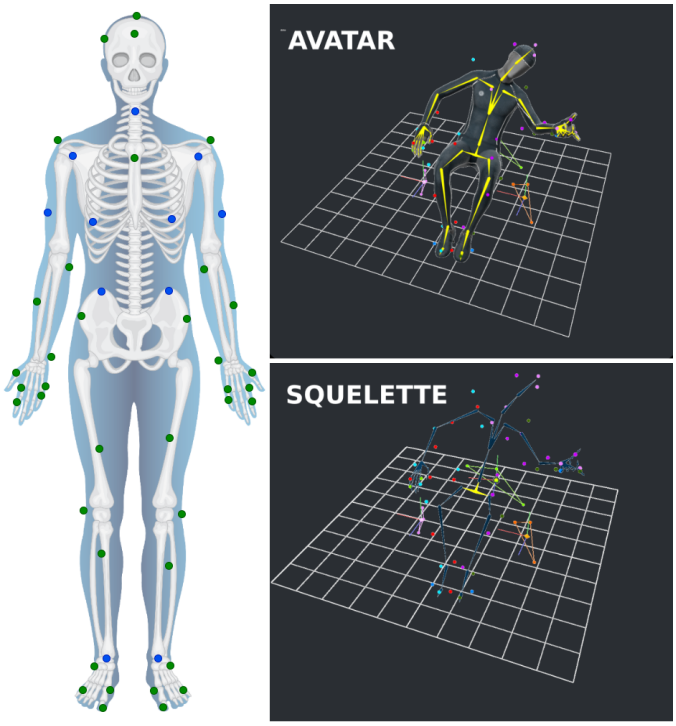

Figure 7.2 – Marker set used and rendering previews in the capture software.

women and 1 man among the participants in real disability situation. We selected among the simulation participants, 2 students from the field APAH (Adapted Physical Activity and Health) that, unlike the other participants, can performs fall scenarios. More details on the participants are given in Table 7.1.

	Handicap	**Actors**	**Total**
Men	1	2	3
Women	2	3	5
Total	3	5	8

Table 7.1 – Details on captured participants.

7.4 Recorded Data

The recorded data were captured from the 8 participants previously described, following the 12 scenarios. Each scenario was executed 1 or 2 times. Following the recordings, the data were post-processed with the *Motive* software.

We managed to obtain approximately **115 minutes** of recording at a sampling rate of 120 Hz. Two types of data were saved after the post-processing: raw data and skeleton data.

7.4.1 Raw Data

Raw data represent recordings of the markers' 3D positions as tracked and captured by the cameras of the *Optitrack* system, subsequently labeled and post-processed in the *Motive* software.

7.4.2 Skeleton Data

Skeletal data represents motion captured according to the skeletal structure of the body reconstructed from marker positions. Indeed, as shown in Figure 7.2, it is possible to use the position of the markers to determine the positions of the rigid segments as well as the joints connecting these segments (using for example inverse kinematics). The data for these rigid bodies is tree-structured to represent the skeleton's joint angles and bone lengths. This representation can be exported in FBX or BVH formats, specially designed to store motion data for use in data-driven animations.

7.5 Animation of Virtual Characters for Handi-Motion Database Generation

In order to experiment motion reconstruction with AI models on movements in disability situations, we have chosen to generate video data on which we can run the models. Our approach uses the motion data we previously recorded with the MoCap system to animate avatars in a virtual environment, and then to synthesize videos from the animations.

We propose to generate these videos in *Unity Development Platform* where we create a virtual environment and animate characters. This synthesis approach has many advantages:

— As the environment is virtual, the recording setting can be customized at will. In particular, it is easy to add the desired number of video cameras to capture different angles of view of the motion. Figure 7.3 shows the layout plan of the virtual room.

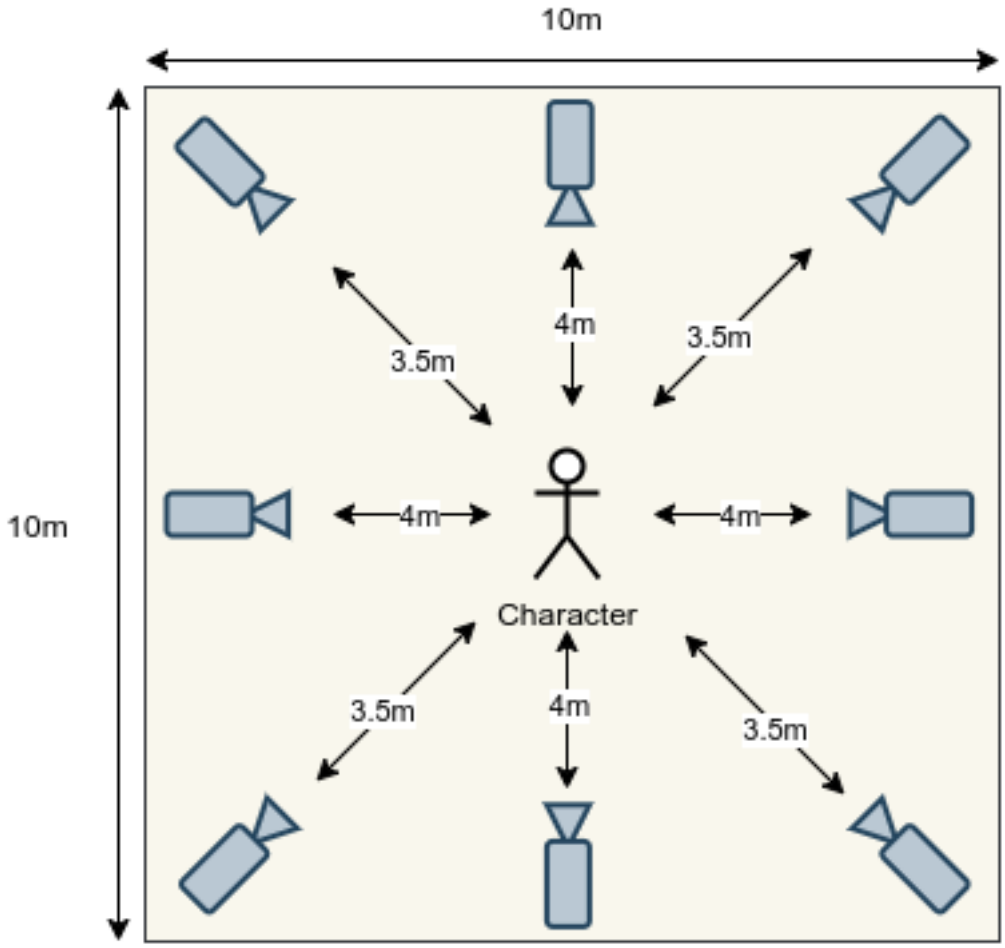

Figure 7.3 – Layout of the virtual environment.

— No post-processing is needed to extract motion data. The extraction can be automatically performed using a script that is executed while an animation is being played. Given that positions and rotations about objects in the scene (including virtual characters) are known in the platform, the script can simultaneously record videos and extract skeleton data.
— It is possible to change the back scene in order to have videos in different environments (See Figure 7.4).

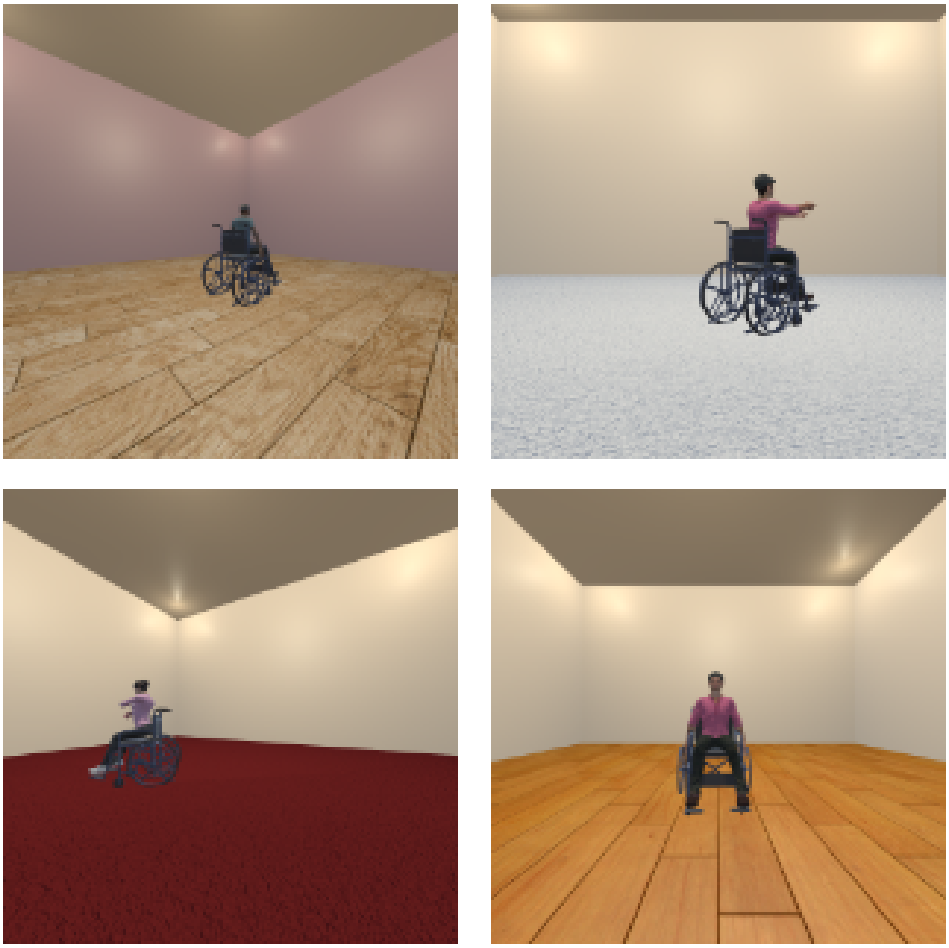

Figure 7.4 – Examples of different back scenes.

— It is also possible to change the virtual character in order to have a variety of morphologies (men, women, tall, short, slim, overweight, etc.). An overview of this aspect is presented in Figure 7.5.

Before proceeding to database generation, we verified that the captured movements were performed without any problem by the virtual characters. We also added a wheelchair

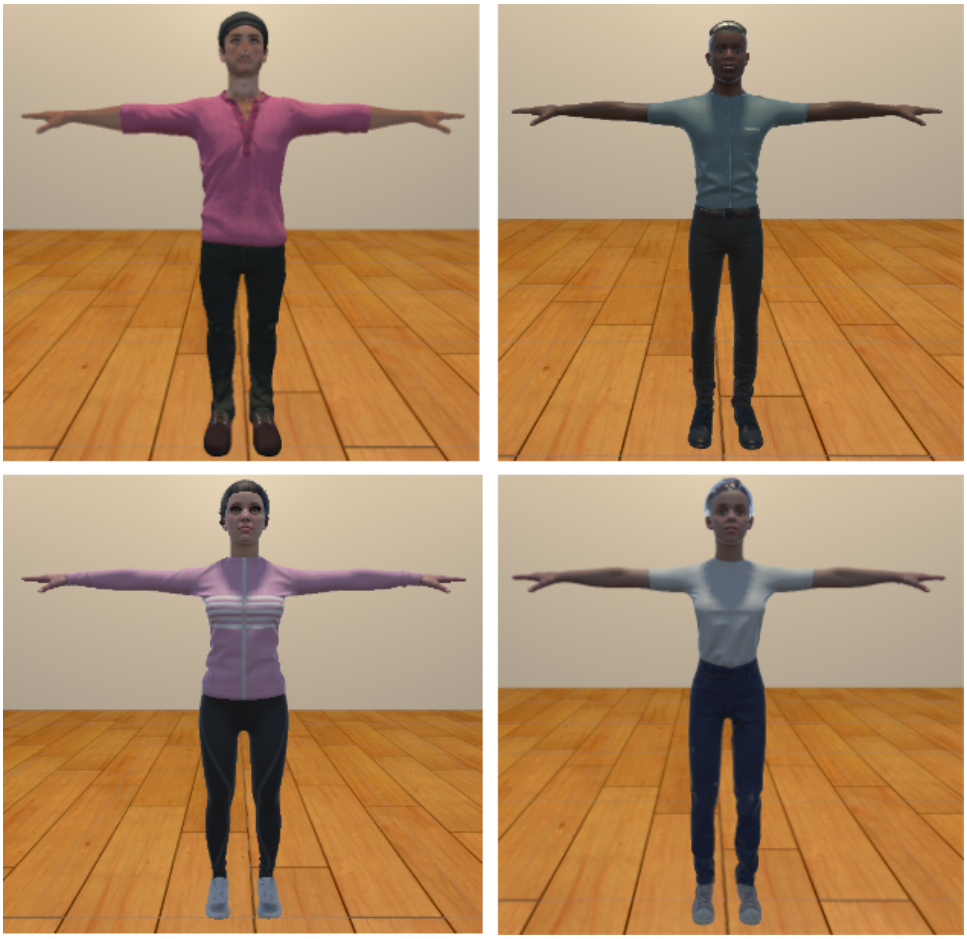

Figure 7.5 – Some of our virtual characters.

(free model available online) and approximately synchronized interactions of the virtual character with it as they have been created as two independent elements in the scene. We faced some difficulties in this synchronization:

— The size and the position of the wheelchair had to be adjusted to match the morphology of the avatar. This action had to be made for each avatar and for each animation.

— During the recording of the *MoCap* data, we recorded the movements of the wheelchair with some markers (4 placed on the back of the seat, and 3 on each wheel's spokes). However, the animation obtained from that (animation of an object with 3 rigid bodies) had to be modified to match the structure of the wheelchair in the virtual environment.

Database Generation

We generated our database by extracting different data from the animations played in the virtual environment. The database contains sets of videos recordings and synchronized 2D/3D skeletal postures.

We performed an automatic extraction of data using a script that runs when an animation is played. Through this script, we produced video data using the cameras in the scene while simultaneously extracting 2D and 3D skeletal postures. To ensure that the videos and skeletal postures are synchronized, we record at each frame of the animation the following data:

1. images of the scene seen by all cameras. With the 8 cameras, we have 8 different images (different viewpoints).
2. the 3D skeletal posture of the character, represented as the set of 3D position coordinates of all joints $\{\{p\}_1, \{p\}_2, \ldots \{p\}_j\}$ with j the number of joints and $\{p\} = [x, y, z]$. This data serves as ground truth data.
3. 2D skeletal postures, one for each camera, by projecting the 3D skeletal posture on their respective screen. This produces a set of 2D pixel locations of each joint, representing the ground truth of 2D skeletal postures for images produced by each camera. With 8 cameras we have 8 sets of 2D joint locations defined by $\{\{p\}_1, \{p\}_2, \ldots \{p\}_j\}$ with j the number of joints and $\{p\} = [x, y]$.

Synchronized skeletal postures and videos are generated by gathering respectively skeletal postures and images in chronological order. For every animation, the generation produces

8 sequences of 2D skeletal postures, 8 videos (sequences of images), and the corresponding sequence of 3D skeletal postures (Figure 7.6).

For the experiment described below, we use our *Handi-Motion* database which contains 60,112 video frames (7,514 per camera), with the corresponding 60,112 2D skeletal postures and 7,514 3D skeletal postures.

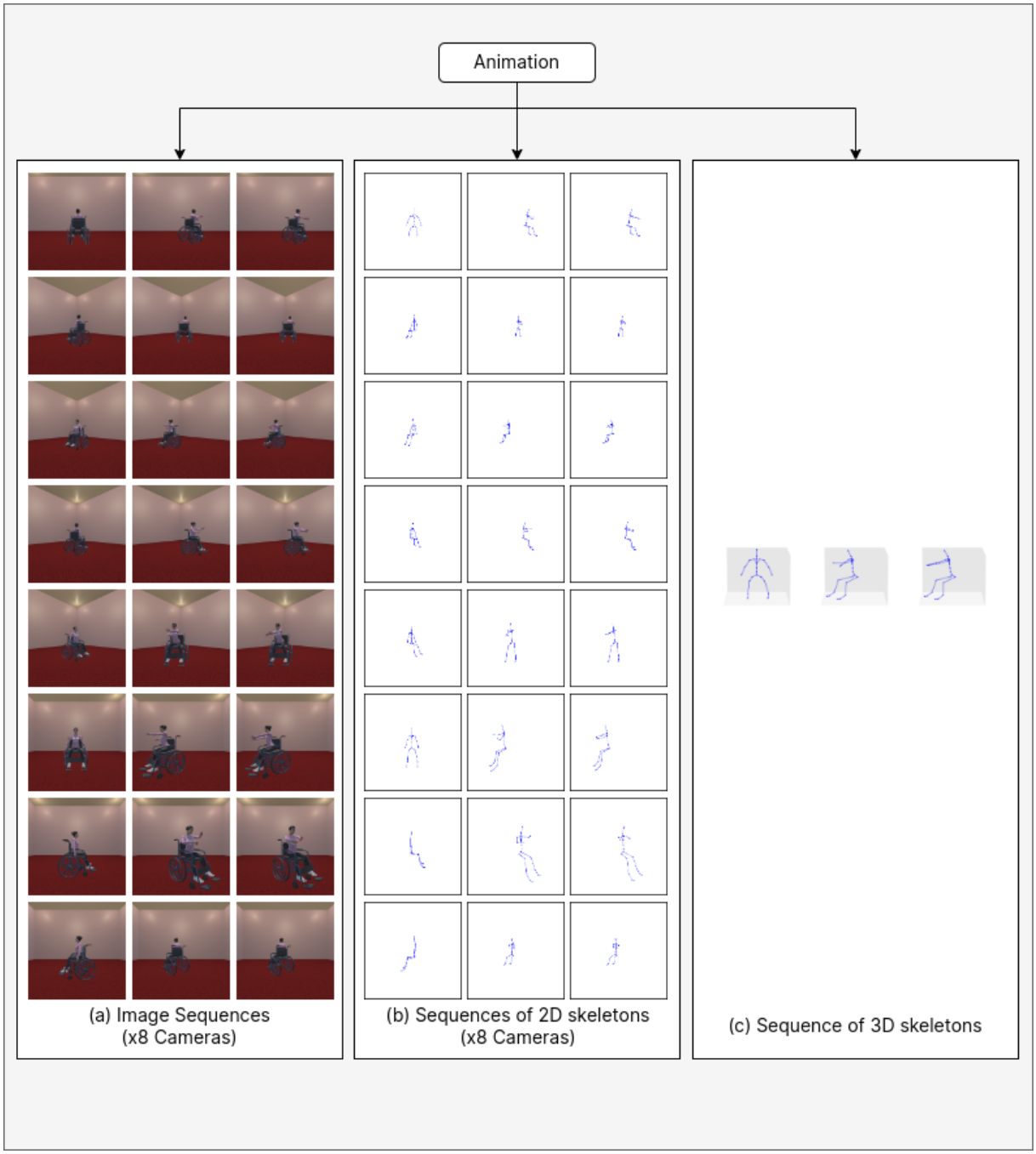

Figure 7.6 – Generation of database content. Representation of data generated for each animation. Left: the sequences of images produced by the 8 cameras (1 row = 1 camera). Middle: Sequences of 2D skeletal postures for each of the 8 cameras. Right: Corresponding sequence of 3D skeletal postures.

7.6 Experiments with Deep Learning

The experiments presented here are realized on the *Handi-Motion* database of animated videos previously presented. We want to test the efficiency of solutions that we presented in Chapter 6, trained on the massive database *Human3.6m*.

Our motion reconstruction solution consists of i) estimating a sequence of 3D skeletal postures from a sequence of 2D poses (using a STAR estimator), then ii) reconstructing the motion with our correction algorithm. The full process is presented in Figure 7.7 and, as shown in the figure, we reconstruct motion from a sequence of 2D poses previously estimated from the video. Therefore, in our experiment presented in this section, we reconstruct motion in 3D from ground truth (GT) sequences of 2D poses.

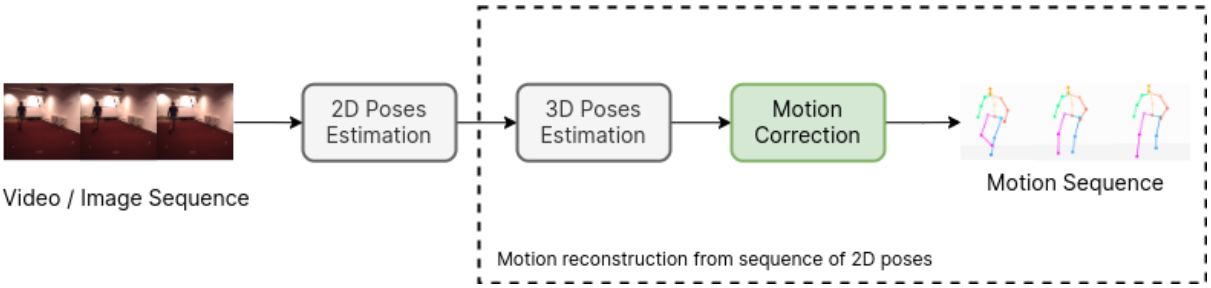

Figure 7.7 – Pipeline of the motion reconstruction solution.

Figure 7.8 shows the context in which the experiment is conducted. In this context, we used the reconstruction models that where originally trained with the database *Human3.6m*. We evaluate the performance of these models on *Handi-Motion* database. The criteria of evaluation consist of the performance on spatial accuracy and temporal consistency with metrics on joint positions (MPJPE), velocity (MPJVE), acceleration (MPJAccE), as well as bone lengths (MBLE). We also evaluate the skeletal consistency in the reconstruction process with the metrics SVE proposed in Chapter 6. We present the results obtained using the STAR models previously defined, namely PoseFormer [140], MotioNet [102], and AANet [17] for the estimation of sequences of 3D poses.

Two experimentation results are presented. Firstly, we present evaluation results for each model on the database while comparing results of 3D poses estimation and 3D poses estimation with motion correction. Secondly, we compare the performance of the pipeline (3D poses estimation + correction) on *Human3.6m* database with that on *Handi-Motion* database.

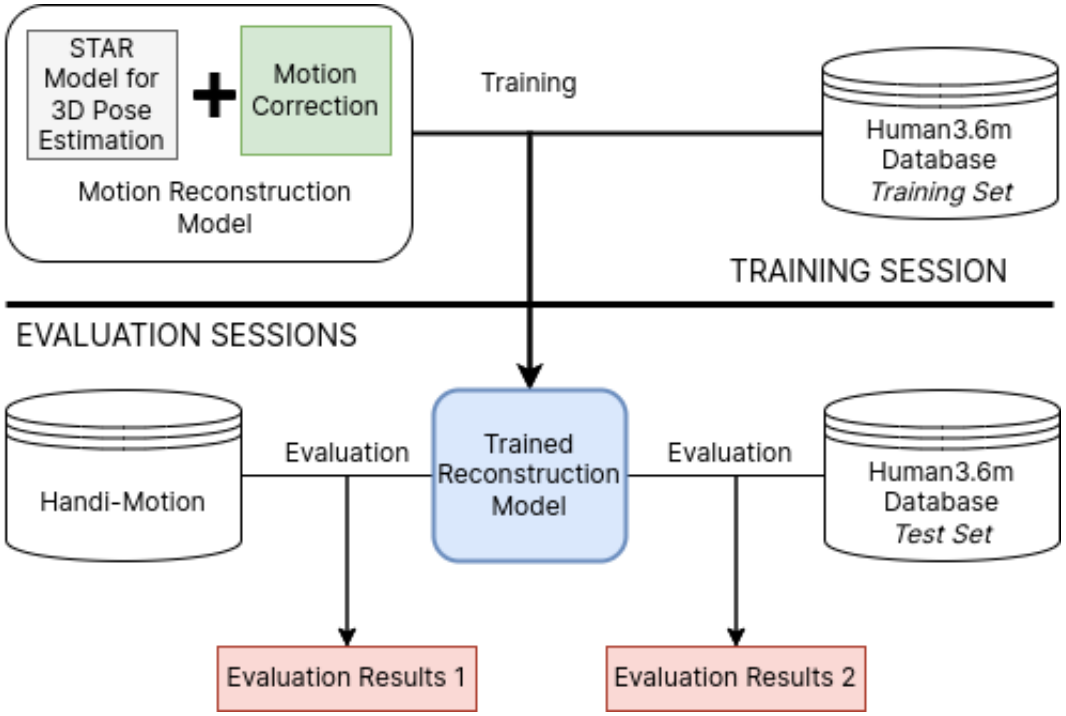

Figure 7.8 – Context of experimentation. Models are trained on part of Human3.6m database. Evaluations are performed on the remaining part of Human3.6m database and Handi-Motion database.

7.6.1 Comparative Results with Motion Correction

	MPJPE	MPJVE	MPJAccE	MBLE	SVE
STAR model	125.21	5.47	2.3	**34.69**	0
STAR model + Correction	**124.11**	**5.31**	**2.07**	36.54	0

(a) Results with MotioNet [102]

	MPJPE	MPJVE	MPJAccE	MBLE	SVE
STAR model	91.72	**3.3**	1.39	**31.85**	8.41
STAR model + Correction	**91.71**	3.44	1.39	36.54	**0**

(b) Results with PoseFormer [140]

	MPJPE	MPJVE	MPJAccE	MBLE	SVE
STAR model	**80.24**	**3.88**	1.77	**33.38**	2.29
STAR model + Correction	83.12	3.94	**1.64**	36.54	**0**

(c) Results with AANet [17]

Table 7.2 – Quantitative results of motion reconstruction on Handi-Motion database. Errors are in mm.

In Table 7.2, we present the results on evaluation of the STAR models before and after motion correction to prove that our algorithm improves the quality of the reconstruction. The results obtained in this experiment are similar to those of the ablation study in Chapter 6. The motion correction system mostly improves temporal metrics, as shown by the reduction in velocity and acceleration errors (respectively MPJVE and MPJAccE metrics), while showing little improvement in MPJPE (except for the STAR model AANet [17]). The results are more noticeable on the acceleration descriptor for which there is improvement regardless of the STAR model used. There is a little improvement on the velocity for the model MotioNet [102] but a slight increase in error for the others. Bone length errors (MBLE), contrary to previous results, increase slightly after correction but are still in the same order of magnitude. As these errors are related to the neural network that estimates bone lengths, improving this model later will produce better results.

7.6.2 Comparative Results between Databases

Here we compare the results of our complete reconstruction pipeline (3D estimation + correction) obtained on the original database (Human3.6m [52]) that was used to train the

neural network models, with the results obtained when using these same models on the database *Handi-Motion*. Comparative results are presented in Table 7.3, while Figure 7.9 and Figure 7.10 show examples of visualization.

Database	MPJPE	MPJVE	MPJAccE	MBLE
Human3.6m [52]	52.85	2.73	1.22	3.76
Handi-Motion	124.11	5.31	2.07	36.54

(a) MotioNet [102]

Database	MPJPE	MPJVE	MPJAccE	MBLE
Human3.6m [52]	29.71	1.34	0.76	3.76
Handi-Motion	91.71	3.44	1.39	36.54

(b) PoseFormer [140]

Database	MPJPE	MPJVE	MPJAccE	MBLE
Human3.6m [52]	44.63	2.27	1.00	3.76
Handi-Motion	83.12	3.94	1.64	36.54

(c) AANet [17]

Table 7.3 – Quantitative results of the motion reconstruction pipeline (3D pose estimation + correction) on our Handi-Motion database compared to Human3.6m [52].

As may be seen from the results in Table 7.3, the reconstruction on our new database is less accurate on joint positions, approximately 2-3 times. The temporal quality of reconstructed motions is reduced as well.

These results are induced by significant errors of more than 3cm in estimating bone lengths, i.e. 10 times the errors on Human3.6m dataset. An observation of visual results in Figure 7.9 confirms this interpretation. The main reason for this increase in errors lies in the differences between the two databases, Human3.6m and Hand-Motion, due to the different environments in which the videos were recorded. This includes the distance between the camera and the MoCap actor. We were able to verify this by computing a scale factor between the 3D skeletal structure of people in *Human3.6m* database and in our *Handi-Motion* database. We found that the scaling factor is approximately 1.005, meaning that the skeletal structures are similar, therefore these errors come from 2D information (video). To further confirm our hypothesis, we analyzed results obtained when moving the cameras closer to or further from the virtual character for the same movement. We observed that error on bone lengths varies accordingly.

In addition, after visualizing the reconstructed movements, we found that gestures are half-executed, i.e. the model is not able to reproduce complete gestures accurately.

For example, arms are halfway stretched when they should be completely stretched, or halfway bent when they should be fully bent. This greatly impacts both the temporal factors (velocity and acceleration) and the spatial accuracy. This can be explained by the difference between the two databases (*Human3.6m* and *Handi-Motion*) in terms of categories of movements. Indeed, models were previously trained and evaluated with 2 sets of data from the same database, and so, with movements of the same categories. As a result, their performance is not reflected in the same way when it comes to movements of new categories.

The visualization in Figure 7.9 shows that the system still manages to reconstruct skeletal postures that tend to follow the original movement, even if not accurately.

From these results, we can conclude that existing DL solutions for motion reconstruction are relatively dependent on the database on which they were trained, which reduces their generative capacity. In other words, their performance is reduced on databases different from their training database.

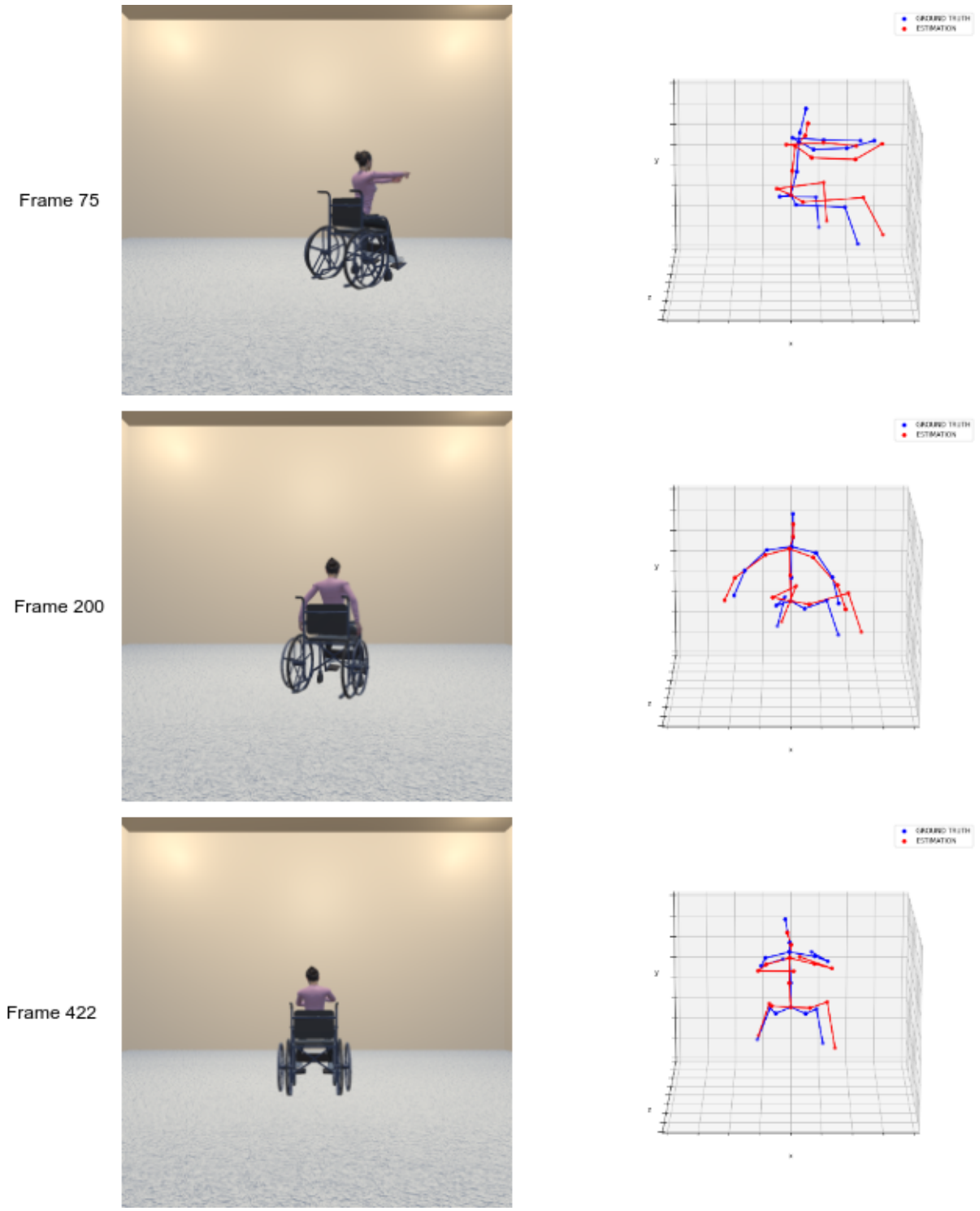

Figure 7.9 – First example of visual results. In blue the original pose and in red the reconstructed pose. Back View Camera.

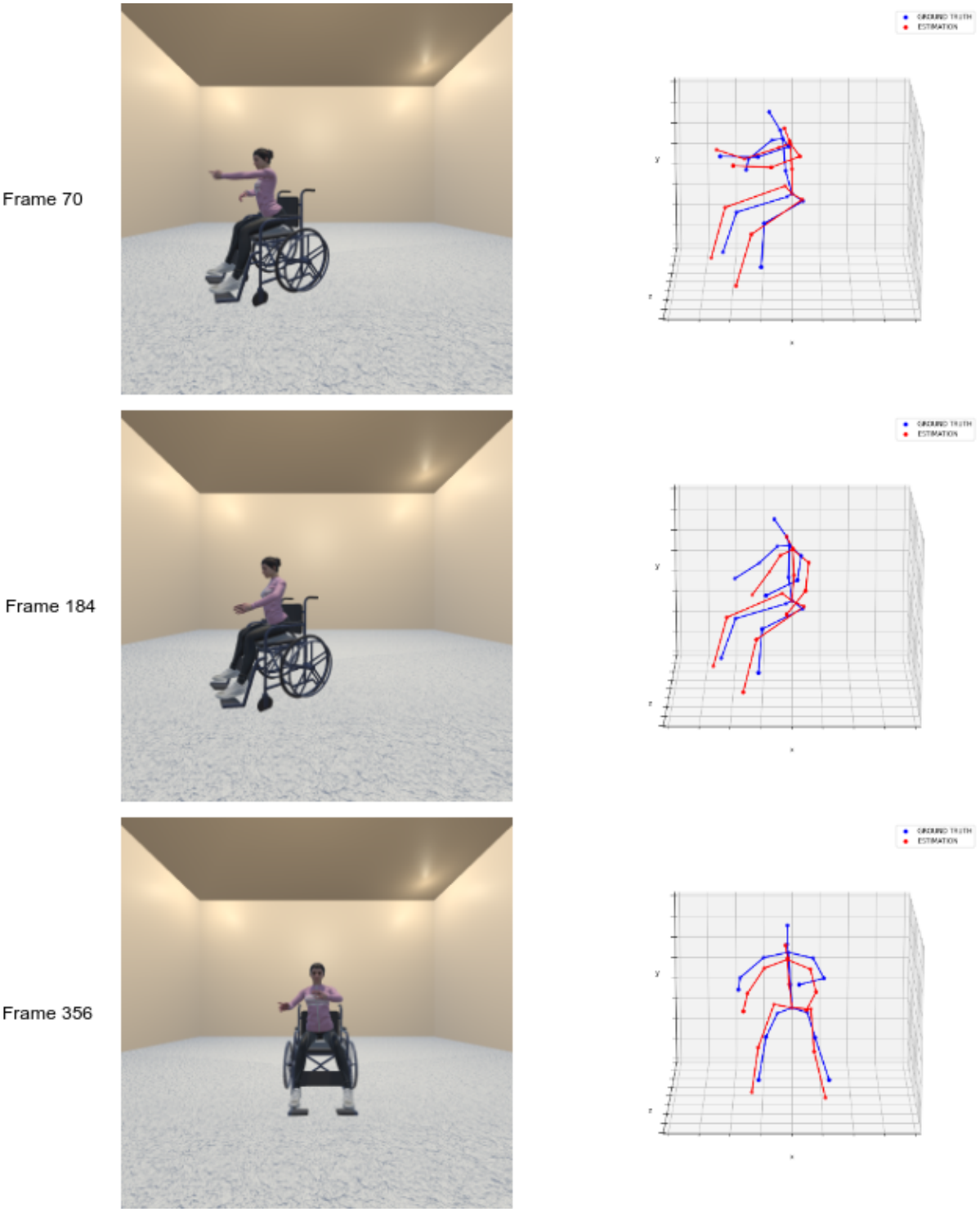

Figure 7.10 – Second example of visual results. In blue the original pose and in red the reconstructed pose. Front View Camera.

7.7 Conclusion

In this chapter, we first presented our new Handi-Motion database containing motion from people with disabilities. We then experimented our motion reconstruction models on these data.

To overcome the lack of data on motor disability movements, we decided to build our own database. We first captured and recorded data using an optical *MoCap* systems. In order to augment our data and to experiment motion reconstruction from video with AI models, we generated animated videos from the 3D captured motions with *Unity Development Platform*. An experimental version of the generation process shows that the virtual characters perform without difficulties the captured motion. We also manage to add a wheelchair to the scene and synchronized it with the position of the character.

We tested our previously developed motion reconstruction models on the generated videos. The reconstruction models that were trained with the database *Human3.6m* [52]) achieved at best an accuracy approximately 2 times less well on our *Handi-Motion* database. Although they were not highly accurate, they managed to recover a visually acceptable motion. This confirms that our approach to generate a database of movements in disability situations is needed in order to design adapted AI models for motion reconstruction from video when developing related applications.

During our experiments, it was possible to fine-tune our existing models on the Handi-Motion database, that is, to retrain them with data from this database. However, we chose not to do so for the time being because we believed that fine-tuning the models will most likely yield better evaluation results, in particular the neural network for bone length estimation whose improvement will definitely increase the accuracy of the reconstruction system. Indeed, we wanted to focus first on the efficiency of our motion reconstruction method on new data, especially the motion correction system. Moreover, fine-tuning a model does not guarantee that the models will have better generative capacity as it may be over-fitted on the new data. Fine-tuning experiments will be carried out in future work.

There is still work to be done on this database. We plan to carry out a perceptual evaluation of the animated videos to validate this database. Before that, we need to improve the quality of the virtual character, as well as the animation of the wheelchair (better synchronization with the character, wheels that turn according to the movement). We then want to evaluate the movements from these videos in terms of their degree of naturalness. We also intend to carry out an analytical comparison between motion

originally rendered by the *MoCap* system and those performed by the virtual character.

Chapter 8

Conclusion

Contents

8.1 Contributions

The goal of this thesis was to propose solutions to the problem of 3D motion reconstruction from videos in complex situations. More specifically, our objective was to propose an alternative solution to motion capture systems which are difficult to use in case of motor disability. We thus provided the basis for designing tools for applications that improve the daily lives of disabled people. We proposed to use deep learning methods from Computer Vision in the context of human pose estimation from images, to produce motion data that can be used for various tasks such as motion analysis, synthesis, etc. Therefore, we needed to reconstruct motion as a sequence of postures with the following characteristics: a good spatial accuracy while maintaining a stable skeletal structure and ensuring the temporal continuity between the postures.

We addressed this issue with two goals: motion reconstruction from video in general and its use in the specific case of motor disability movements. For motion reconstruction, we proposed a 2-steps approach that consists of (i) estimating motion as a sequence of 3D skeletal postures, (ii) reconstructing motion of better quality with a correction algorithm. For the specific case of motor disability, we experimented our motion reconstruction solution on a database that we designed and recorded in real and simulated motor disability conditions.

Motion reconstruction from video with Deep Learning

Existing work on motion reconstruction from video using deep learning techniques started with human pose estimation. In this field, motion is represented as a sequence of postures of the human skeleton, each posture being defined as a set of positions of the skeletal joints. Methods for motion reconstruction from video generally comes from Computer Vision. After studying various approaches, we decided to learn how to transform a sequence of 2D postures into a sequence of 3D postures using deep learning techniques on sequential data such as recurrent neural networks or temporal convolution networks. The whole reconstruction process would normally consist in estimating 2D postures from video, then transform them into 3D postures. But, thanks to the great performances achieved in 2D pose estimation, we decided to focus on the transformation of skeletal postures from 2D to 3D. We can use existing 2D pose estimator in the reconstruction process from video.

Our first experiments on neural architectures, aiming at learning the transformation of a sequence of 2D postures into a sequence of 3D postures by exploiting temporal convolution, followed the traditional approach used in computer vision, with a learning process based on minimizing spatial errors in joint positions. However, we found that this was not sufficient to guarantee the quality of the reconstructed movement, and in particular this did not respect temporal consistency.

Therefore, we decided to jointly learn a spatiotemporal representation of the motion. To this end, we chose to represent the motion as a spatiotemporal graph 3D+t that connects the skeletal postures. We then applied the discrete Laplacian operator on the graph to extract Laplacian coordinates that encode spatial and temporal relations at each joint level. From this representation, we defined the *Laplacian Loss* function to train the neural network. We proved that training a DNN with a combination of both the *joint positions loss* and the *Laplacian Loss*, improves the quality of the reconstructed motion.

Following the results of this first experiment, and with the goal of preparing for real-time usage of a motion reconstruction system from video, we designed a *lightweight neural network* for sequence-to-sequence poses estimation.

Subsequently, we designed the second part of our methodology, which aims to reconstruct better quality movements from estimated sequences of poses. We found that none of the existing approaches was able to achieve results in all aspects of motion quality, including spatial accuracy, temporal consistency and preservation of skeletal structure.

We have therefore proposed a motion correction system *MoCoSys* that complements

existing solutions on the aspects where they are lacking. This system uses DL techniques in Laplacian coordinates space to improve the spatiotemporal quality of the sequence of postures. At the same time, it uses an algorithm to adjust the distance between the skeletal joints for each posture, in order to ensure that the skeletal structure is preserved. Results from a comparative evaluation have proved the efficiency of the approach.

Handi-Motion database

We chose to apply our motion reconstruction system to motor disability situations. The final aim of our thesis was to propose a system that can facilitate the development of applications to ease the daily life and promote autonomy of people with motor disability. Applications may need to track and capture motion data for various tasks such as motion and behavior analysis, activity and gesture recognition, or use it for tasks such as motion synthesis.

However, we realized that there was no digital recording of motion data in motor disability situations. We believe this is mainly due to the difficulty of using MoCap systems in such situations, for a variety of reasons, including constraints on use, the existence of wheelchair occlusions for systems based on optical markers, and so on. Video-based motion reconstruction systems that use deep learning techniques to develop and train artificial intelligence models should be a solution to record such data. However, these models need existing data to learn before they can perform accurate reconstruction.

We decided to collect the data we needed by first using a MoCap optical system to capture a small amount of high-resolution data from a set of people. Among them, there were people with disabilities who were able to satisfy the constraints required by the system and people simulating disability situations. Then, we used the captured data to generate synthesized videos from animated virtual characters. With this synthesis approach, we were able to generate a much larger amount of data than the initial MoCap data, allowing us to evaluate AI models for motion reconstruction.

Using this database, we carried out an experiment to evaluate our previous motion reconstruction models presented in Chapter 6. This experiment confirmed the need to create this database, as existing models were less accurate when reconstructing motion different from their training videos. Nevertheless, they succeeded in rendering a visually acceptable reconstructed motion.

The results obtained in this experiment enable us to validate the importance of this database and to discover new research perspectives.

8.2 Perspectives

We present here various perspectives of research for the continuation of this work.

Evaluating Handi-Motion database

Our *Handi-Motion* database contains motion data of a specific type, usually performed by people in disability situations. The participants to the recording sessions were a group of people with some of them in actual disability situations and the others, people simulating these situations. We asked the participants to play daily life activities. We then used a data-driven 3D animation program to produce videos of movements performed by virtual characters. By using virtual characters, this approach made it possible to produce anonymous videos and protect the privacy of the participants in the MoCap sessions. The videos can then be showed to a wider audience.

As the videos were synthesized, it is of paramount importance that people of the motor disability community validate this database. Moreover, it is necessary to evaluate how these movements, performed in a virtual environment, can be perceived by a representative community familiar with motor disability, including people with disability, occupational therapists, medical experts, etc.

There are various aspects to validate in the database. The first one is related to motion of actors that simulate a disability situation. We have to assess first if people are capable of differentiating movements of people in actual disability situation from those of people who simulate. The second aspect is related to the fact that videos are synthesized. According to this point, we can evaluate the level of acceptance of this type of videos in the community of people with disabilities. In particular, it can be interesting to evaluate various aspects of the animations, by varying the motion representation (points or skeletal displays, appearance of the avatar, point of view, etc).

In addition to these perceptual evaluations, we think it would be interesting to assess the potential effect that the re-targeting on avatar has on the MoCap animations.

Performing further tests of motion reconstruction on Handi-Motion database

During this thesis, we experimented our motion reconstruction model from generated videos of *Handi-Motion* database. We used for this experiment ground truth sequences

of 2D poses, as our reconstruction model operates on sequences of estimated 2D poses. There are therefore some additional tests to perform:

1. **Testing 2D pose estimation.** We need to evaluate the performance of 2D pose estimation. This can allow us to verify if the existing estimators such as OpenPose [13] or HRNet [110] are robust to situations where parts of the body are hidden by the wheelchair. Indeed, these situations can lead to errors in estimating 2D joints locations, leading in turn to errors in the 3D skeletal postures estimation.
2. **Testing the robustness of our motion reconstruction system.** We believe that our correction system can partially fix some errors in 2D poses obtained from 2D poses estimators, thanks to the spatiotemporal connection within the Laplacian modeling. We intend to verify this by generating some examples of situations with controlled artefacts and by observing how the model performs.

Improving the Algorithm of the Motion Correction System

The neural network of our motion correction system that estimates skeletal structure (i.e., length on the bones) from the sequence of 2D poses, is less accurate on *Handi-Motion* database. Indeed, with *Human3.6m* [52], the database used to train it, it produces skeletal structure with an approximate error of 3mm. However, this error rises to 36mm on *Handi-Motion* database, about 10 times greater. Although fine-tuning the model on *Handi-Motion* (that is retraining the model on the current database) will improve the results on this specific database, it would be more interesting to develop a new model capable of estimating skeletal structure on different databases with better accuracy. A solution can be to design the model to estimate the bone lengths from the video instead of the sequence of 2D poses. Another alternative for improvement would be to train the current neural network on various other databases in order to increase it generalization capacity.

Improving the Lightweight Model for Motion Reconstruction

In Chapter 5, we proposed a lightweight neural network to estimate sequences of 3D skeletal postures from sequences of 2D poses. As part of our original objective, and for future development of real-time applications, it would be interesting to further improve this solution, building in this way the complete pipeline of the motion reconstruction solution.

In addition, DL models are usually trained on massive amount of data, making their performance closely related to databases used to train them. This creates models with less capacity for generalization and lower performance on new types of data. To ensure that the model can be used in a wide variety of situations, it will be interesting to try frugal AI approaches that learn on small amount of data.

Applying the Motion Correction Algorithm to a Wider Range of Motion

Similar to the context of motor disability, sign language also finds itself in a situation where there is a lack of movement data. Therefore, there is a need to use generative AI models to produce more data with body, hand and face reconstruction [134]. However, it will be essential to correct this motion data, especially hand gestures where a small variation can change the semantic, or where a facial expression can negate the sentence. We would like to combine this kind of generative models with an adaptation of our motion correction algorithm to improve the quality of the reconstruction.

BIBLIOGRAPHY

[1] A. Agarwal and B. Triggs, « Recovering 3D Human Pose from Monocular Images », *in*: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.*1* (Jan. 2006), pp. 44–58, DOI: 10.1109/TPAMI.2006.21.

[2] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele, « Monocular 3D Pose Estimation and Tracking by Detection », *in*: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, June 2010, pp. 623–630, DOI: 10.1109/CVPR.2010.5540156.

[3] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele, « Pictorial Structures Revisited: People Detection and Articulated Pose Estimation », *in*: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops), Miami, FL, June 2009, pp. 1014–1021, DOI: 10.1109/CVPR.2009.5206754.

[4] Andreas Aristidou and Joan Lasenby, « FABRIK: A fast, iterative solver for the Inverse Kinematics problem », *in*: *Graph. Models* 73.*5* (Sept. 2011), pp. 243–260, ISSN: 1524-0703, DOI: 10.1016/j.gmod.2011.05.003, URL: http://dx.doi.org/10.1016/j.gmod.2011.05.003.

[5] Anurag Arnab, Carl Doersch, and Andrew Zisserman, « Exploiting Temporal Context for 3D Human Pose Estimation in the Wild », *in*: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, June 2019, pp. 3390–3399, DOI: 10.1109/CVPR.2019.00351.

[6] Bruno Artacho and Andreas Savakis, « UniPose+: A Unified Framework for 2D and 3D Human Pose Estimation in Images and Videos », *in*: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.*12* (Dec. 1, 2022), pp. 9641–9653, DOI: 10.1109/TPAMI.2021.3124736.

[7] Niloofar Azizi et al., « 3D Human Pose Estimation Using Möbius Graph Convolutional Networks », *in*: *Computer Vision – ECCV 2022*, ed. by Shai Avidan et al., vol. 13661, Cham, 2022, pp. 160–178, DOI: 10.1007/978-3-031-19769-7_10.

[8] Martin Bergtholdt et al., « A Study of Parts-Based Object Class Detection Using Complete Graphs », *in*: *International Journal of Computer Vision* 87.*1-2* (Mar. 2010), pp. 93–117, DOI: 10.1007/s11263-009-0209-1.

[9] Sandika Biswas et al., « Lifting 2d Human Pose to 3d : A Weakly Supervised Approach », *in*: *International Joint Conference on Neural Networks, IJCNN 2019*, May 3, 2019, pp. 1–9, arXiv: 1905.01047, URL: http://arxiv.org/abs/1905.01047.

[10] Arij Bouazizi, Ulrich Kressel, and Vasileios Belagiannis, « Learning Temporal 3D Human Pose Estimation with Pseudo-Labels », *in*: *17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2021, DOI: 10.1109/AVSS52988.2021.9663755, arXiv: 2110.07578 [cs].

[11] Arij Bouazizi et al., « Self-Supervised 3D Human Pose Estimation with Multiple-View Geometry », *in*: *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021), Jodhpur, India, Dec. 15, 2021, pp. 1–8, DOI: 10.1109/FG52635.2021.9667074.

[12] Yujun Cai et al., « Exploiting Spatial-Temporal Relationships for 3D Pose Estimation via Graph Convolutional Networks », *in*: *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-Octob, Oct. 1, 2019, pp. 2272–2281, DOI: 10.1109/ICCV.2019.00236.

[13] Zhe Cao et al., « OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields », *in*: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.*1* (Jan. 1, 2021), pp. 172–186, DOI: 10.1109/TPAMI.2019.2929257.

[14] Joao Carreira et al., « Human Pose Estimation with Iterative Error Feedback », *in*: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, June 2016, pp. 4733–4742, DOI: 10.1109/CVPR.2016.512.

[15] Pamela Carreno-Medrano et al., « Corpus Creation and Perceptual Evaluation of Expressive Theatrical Gestures », *in*: *Intelligent Virtual Agents*, ed. by Timothy Bickmore, Stacy Marsella, and Candace Sidner, vol. 8637, Cham, 2014, pp. 109–119, DOI: 10.1007/978-3-319-09767-1_14.

[16] Ching-Hang Chen et al., « Unsupervised 3D Pose Estimation With Geometric Self-Supervision », *in*: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, June 2019, pp. 5707–5717, DOI: 10.1109/CVPR.2019.00586.

[17] Tianlang Chen et al., « Anatomy-Aware 3D Human Pose Estimation With Bone-Based Pose Decomposition », *in*: *IEEE Transactions on Circuits and Systems for Video Technology* 32.*1* (1 Jan. 2022), pp. 198–209, DOI: 10.1109/TCSVT.2021.3057267.

[18] Yilun Chen et al., « Cascaded Pyramid Network for Multi-person Pose Estimation », *in*: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, June 2018, pp. 7103–7112, DOI: 10.1109/CVPR.2018.00742.

[19] Yucheng Chen, Yingli Tian, and Mingyi He, « Monocular Human Pose Estimation: A Survey of Deep Learning-Based Methods », *in*: *Computer Vision and Image Understanding* 192 (Mar. 2020), p. 102897, DOI: 10.1016/j.cviu.2019.102897.

[20] Zhangmeng Chen et al., « DGFormer: Dynamic Graph Transformer for 3D Human Pose Estimation », *in*: *Pattern Recognition* 152 (Aug. 2024), p. 110446, DOI: 10.1016/j.patcog.2024.110446.

[21] Jia-Ching Cheng and J.M.F. Moura, « Capture and Representation of Human Walking in Live Video Sequences », *in*: *IEEE Transactions on Multimedia* 1.*2* (June 1999), pp. 144–156, DOI: 10.1109/6046.766736.

[22] Hongsuk Choi et al., « Beyond Static Features for Temporally Consistent 3D Human Pose and Shape from a Video », *in*: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 1964–1973, DOI: 10.1109/CVPR46437.2021.00200.

[23] Sungho Chun, Sungbum Park, and Ju Yong Chang, « Learnable Human Mesh Triangulation for 3D Human Pose and Shape Estimation », *in*: *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, Jan. 2023, pp. 2849–2858, DOI: 10.1109/WACV56688.2023.00287.

[24] Hai Ci et al., « Optimizing Network Structure for 3D Human Pose Estimation », *in*: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), Oct. 2019, pp. 2262–2271, DOI: 10.1109/ICCV.2019.00235.

[25] Grazia Cicirelli et al., « Human Gait Analysis in Neurodegenerative Diseases: A Review », *in*: *IEEE Journal of Biomedical and Health Informatics* 26.*1* (Jan. 2022), pp. 229–242, DOI: 10.1109/JBHI.2021.3092875.

[26] *CMU Graphics Lab Motion Capture Database*, http://mocap.cs.cmu.edu/.

[27] Rachel E Cowan et al., « Recent Trends in Assistive Technology for Mobility », *in*: *Journal of NeuroEngineering and Rehabilitation* 9.*1* (2012), p. 20, DOI: 10.1186/1743-0003-9-20.

[28] Rishabh Dabral et al., « Learning 3D Human Pose from Structure and Motion », *in*: *Computer Vision – ECCV 2018*, vol. 11213 LNCS, 2018, pp. 679–696, DOI: 10.1007/978-3-030-01240-3_41.

[29] Rishabh Dabral et al., « Multi-Person 3D Human Pose Estimation from Monocular Images », *in*: *2019 International Conference on 3D Vision (3DV)*, 2019 International Conference on 3D Vision (3DV), Québec City, QC, Canada, Sept. 2019, pp. 405–414, DOI: 10.1109/3DV.2019.00052.

[30] Qi Dang et al., « Deep Learning Based 2D Human Pose Estimation: A Survey », *in*: *Tsinghua Science and Technology* 24.*6* (Dec. 2019), pp. 663–676, DOI: 10.26599/TST.2018.9010100.

[31] Lazzaro Di Biase et al., « Gait Analysis in Parkinson's Disease: An Overview of the Most Accurate Markers for Diagnosis and Symptoms Monitoring », *in*: *Sensors* 20.*12* (June 22, 2020), p. 3529, DOI: 10.3390/s20123529.

[32] Kyle Duarte and Sylvie Gibet, « Heterogeneous Data Sources for Signed Language Analysis and Synthesis: The SignCom Project », *in*: *7th Int. Conf. on Language Resources and Evaluation (LREC 2010)*, vol. 2, pp. 1–8.

[33] A. Elhayek et al., « Efficient ConvNet-based Marker-Less Motion Capture in General Scenes with a Low Number of Cameras », *in*: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, June 2015, pp. 3810–3818, DOI: 10.1109/CVPR.2015.7299005.

[34] A. Elhayek et al., « MARCOnI—ConvNet-Based MARker-Less Motion Capture in Outdoor and Indoor Scenes », *in*: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.*3* (Mar. 1, 2017), pp. 501–514, DOI: 10.1109/TPAMI.2016.2557779.

[35] Hao-Shu Fang et al., « RMPE: Regional Multi-person Pose Estimation », *in*: *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Oct. 2017, pp. 2353–2362, DOI: 10.1109/ICCV.2017.256.

[36] Mihai Fieraru et al., « AIFit: Automatic 3D Human-Interpretable Feedback Models for Fitness Training », *in*: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, June 2021, pp. 9914–9923, DOI: 10.1109/CVPR46437.2021.00979.

[37] Mihai Fieraru et al., « Learning Complex 3D Human Self-Contact », *in*: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, May 18, 2021, pp. 1343–1351, DOI: 10.1609/aaai.v35i2.16223.

[38] Mihai Fieraru et al., « Three-Dimensional Reconstruction of Human Interactions », *in*: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, June 2020, pp. 7212–7221, DOI: 10.1109/CVPR42600.2020.00724.

[39] Mohsen Gholami et al., « Self-Supervised 3D Human Pose Estimation from Video », *in*: *Neurocomputing* 488 (June 2022), pp. 97–106, DOI: 10.1016/j.neucom.2022.02.076.

[40] Saeed Ghorbani et al., « MoVi: A Large Multi-Purpose Human Motion and Video Dataset », *in*: *PLOS ONE* 16.*6* (June 17, 2021), ed. by Peter Andreas Federolf, e0253157, DOI: 10.1371/journal.pone.0253157.

[41] Sylvie Gibet, « Building French Sign Language Motion Capture Corpora for Signing Avatars », *in*: *Workshop on the Representation and Processing of Sign Languages: Involving the Language Community, LREC 2018*, 2018.

[42] Sylvie Gibet et al., « The SignCom System for Data-Driven Animation of Interactive Virtual Signers: Methodology and Evaluation », *in*: *ACM Transactions on Interactive Intelligent Systems* 1.*1* (Oct. 2011), pp. 1–23, DOI: 10.1145/2030365.2030371.

[43] Ikhsanul Habibie et al., « In the Wild Human Pose Estimation Using Explicit 2D Features and Intermediate 3D Representations », *in*: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, DOI: 10.1109/CVPR.2019.01116, arXiv: 1904.03289.

[44] Ruhan He et al., « Monocular 3D Human Pose Estimation Based on Global Temporal-Attentive and Joints-Attention In Video », *in*: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, June 4, 2023, pp. 1–5, DOI: 10.1109/ICASSP49357.2023.10096111.

[45] Alexis Heloir et al., « Temporal Alignment of Communicative Gesture Sequences », *in*: *Computer Animation and Virtual Worlds* 17.*3-4* (July 2006), pp. 347–357, DOI: 10.1002/cav.138.

[46] Chaoqun Hong et al., « Multimodal Deep Autoencoder for Human Pose Recovery », *in*: *IEEE Transactions on Image Processing* 24.*12* (12 Dec. 1, 2015), pp. 5659–5670, DOI: 10.1109/TIP.2015.2487860, pmid: 26452284.

[47] Mir Rayat Imtiaz Hossain and James J. Little, « Exploiting Temporal Information for 3D Human Pose Estimation », *in*: *Computer Vision – ECCV 2018*, ed. by Vittorio Ferrari et al., vol. 11214, Cham, 2018, pp. 69–86, DOI: 10.1007/978-3-030-01249-6_5.

[48] Junxing Hu et al., « Personalized Graph Generation for Monocular 3D Human Pose and Shape Estimation », *in*: *IEEE Transactions on Circuits and Systems for Video Technology* 34.*4* (Apr. 2024), pp. 2399–2413, DOI: 10.1109/TCSVT.2023.3310525.

[49] Wenbo Hu et al., « Conditional Directed Graph Convolution for 3D Human Pose Estimation », *in*: *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21: ACM Multimedia Conference, Virtual Event China, Oct. 17, 2021, pp. 602–611, DOI: 10.1145/3474085.3475219.

[50] Guoliang Hua et al., « Weakly-Supervised 3D Human Pose Estimation With Cross-View U-Shaped Graph Convolutional Network », *in*: *IEEE Transactions on Multimedia* 25 (2023), pp. 1832–1843, DOI: 10.1109/TMM.2022.3171102.

[51] Dong-Hyun Hwang et al., « Lightweight 3D Human Pose Estimation Network Training Using Teacher-Student Learning », *in*: *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, CO, USA, Mar. 2020, pp. 468–477, DOI: 10.1109/WACV45572.2020.9093595.

[52] 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, DOI: 10.1109/TPAMI.2013.248.

[53] Karim Iskakov et al., « Learnable Triangulation of Human Pose », *in*: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), Oct. 2019, pp. 7717–7726, DOI: 10.1109/ICCV.2019.00781.

[54] J. A. García-García J. G. Enríquez Luis M. Soria Morillo and Juan A. Álvarez-García, « Two Decades of Assistive Technologies to Empower People with Disability: A Systematic Mapping Study », *in*: *Disability and Rehabilitation: Assistive Technology* 0.0 (2023), pp. 1–18, DOI: 10.1080/17483107.2023.2263504, eprint: https://doi.org/10.1080/17483107.2023.2263504.

[55] Zhongyu Jiang et al., « Back to Optimization: Diffusion-Based Zero-Shot 3D Human Pose Estimation », *in*: ().

[56] Hanbyul Joo et al., « Panoptic Studio: A Massively Multiview System for Social Motion Capture », *in*: *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, Dec. 2015, pp. 3334–3342, DOI: 10.1109/ICCV.2015.381.

[57] Thomas N Kipf and Max Welling, « SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS », *in*: *International Conference on Learning Representations (ICLR)*, 2017.

[58] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black, « VIBE: Video Inference for Human Body Pose and Shape Estimation », *in*: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, June 2020, pp. 5252–5262, DOI: 10.1109/CVPR42600.2020.00530.

[59] Franziska Krebs et al., « The KIT Bimanual Manipulation Dataset », *in*: *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, 2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids), Munich, Germany, July 19, 2021, pp. 499–506, DOI: 10.1109/HUMANOIDS47582.2021.9555788.

[60] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi, « PifPaf: Composite Fields for Human Pose Estimation », *in*: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, June 2019, pp. 11969–11978, DOI: 10.1109/CVPR.2019.01225.

[61] Jogendra Nath Kundu et al., « Uncertainty-Aware Adaptation for Self-Supervised 3D Human Pose Estimation », *in*: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, June 2022, pp. 20416–20427, DOI: 10.1109/CVPR52688.2022.01980.

[62] Huaijing Lai, Zhenhua Tang, and Xiaoyan Zhang, « RepEPnP: Weakly Supervised 3D Human Pose Estimation with EPnP Algorithm », *in*: *2023 International Joint Conference on Neural Networks (IJCNN)*, 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, June 18, 2023, pp. 1–8, DOI: 10.1109/IJCNN54540.2023.10191300.

[63] T. Le Naour, N. Courty, and S. Gibet, « Spatiotemporal Coupling with the 3D+t Motion Laplacian », *in*: *Computer Animation and Virtual Worlds* 24.*3-4* (May 2013), pp. 419–428, DOI: 10.1002/cav.1518.

[64] Fran¸cois Lefebvre-Albaret et al., « Overview of the Sign3D Project High-fidelity 3D Recording, Indexing and Editing of French Sign Language Content », *in*: *Third International Symposium on Sign Language Translation and Avatar Technology (SLTAT)*, 2013.

[65] Han Li et al., « Pose-Oriented Transformer with Uncertainty-Guided Refinement for 2D-to-3D Human Pose Estimation », *in*: *Proceedings of the AAAI Conference on Artificial Intelligence* 37.*1* (June 26, 2023), pp. 1296–1304, DOI: 10.1609/aaai.v37i1.25213.

[66] Muyu Li et al., « TSwinPose: Enhanced Monocular 3D Human Pose Estimation with JointFlow », *in*: *Expert Systems with Applications* 249 (Sept. 2024), p. 123545, DOI: 10.1016/j.eswa.2024.123545.

[67] Wenhao Li et al., « Exploiting Temporal Contexts With Strided Transformer for 3D Human Pose Estimation », *in*: *IEEE Transactions on Multimedia* 25 (2023), pp. 1282–1293, DOI: 10.1109/TMM.2022.3141231.

[68] Jiayao Liang and Mengxiao Yin, « SCGFormer: Semantic Chebyshev Graph Convolution Transformer for 3D Human Pose Estimation », *in*: *Applied Sciences* 14.*4* (2024), DOI: 10.3390/app14041646.

[69] Ruixu Liu et al., « Attention Mechanism Exploits Temporal Contexts: Real-Time 3D Human Pose Reconstruction », *in*: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, June 2020, pp. 5063–5072, DOI: 10.1109/CVPR42600.2020.00511.

[70] Matthew Loper et al., « SMPL: A Skinned Multi-Person Linear Model », *in*: *ACM Transactions on Graphics* 34.*6* (Nov. 4, 2015), pp. 1–16, DOI: 10.1145/2816795.2818013.

[71] Sebastian Lutz et al., « Jointformer: Single-Frame Lifting Transformer with Error Prediction and Refinement for 3D Human Pose Estimation », *in*: *2022 26th International Conference on Pattern Recognition (ICPR)*, 2022 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, Aug. 21, 2022, pp. 1156–1163, DOI: 10.1109/ICPR56361.2022.9956366.

[72] Diogo C. Luvizon, David Picard, and Hedi Tabia, « 2D/3D Pose Estimation and Action Recognition Using Multitask Deep Learning », *in*: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2018), pp. 5137–5146, DOI: 10.1109/CVPR.2018.00539, arXiv: 1802.09232.

[73] Naureen Mahmood et al., « AMASS: Archive of Motion Capture As Surface Shapes », *in*: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), Oct. 2019, pp. 5441–5450, DOI: 10.1109/ICCV.2019.00554.

[74] Christian Mandery et al., « The KIT Whole-Body Human Motion Database », *in*: *2015 International Conference on Advanced Robotics (ICAR)*, 2015 International Conference on Advanced Robotics (ICAR), Istanbul, Turkey, July 2015, pp. 329–336, DOI: 10.1109/ICAR.2015.7251476.

[75] Julieta Martinez et al., « A Simple Yet Effective Baseline for 3d Human Pose Estimation », *in*: *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Oct. 2017, pp. 2659–2668, DOI: 10.1109/ICCV.2017.288.

[76] Dushyant Mehta et al., « Monocular 3D Human Pose Estimation in the Wild Using Improved CNN Supervision », *in*: *2017 International Conference on 3D Vision (3DV)*, 2017 International Conference on 3D Vision (3DV), Qingdao, Oct. 2017, pp. 506–516, DOI: 10.1109/3DV.2017.00064.

[77] Dushyant Mehta et al., « Single-Shot Multi-person 3D Pose Estimation from Monocular RGB », *in*: *2018 International Conference on 3D Vision (3DV)*, 2018 International Conference on 3D Vision (3DV), Verona, Sept. 2018, pp. 120–130, DOI: 10.1109/3DV.2018.00024.

[78] Dushyant Mehta et al., « VNect: Real-Time 3D Human Pose Estimation with a Single RGB Camera », *in*: *ACM Transactions on Graphics* 36.*4* (Aug. 31, 2017), pp. 1–14, DOI: 10.1145/3072959.3073596.

[79] Dushyant Mehta et al., « XNect: Real-time Multi-Person 3D Motion Capture with a Single RGB Camera », *in*: *ACM Transactions on Graphics* 39.*4* (4 2020), DOI: 10.1145/3386569.3392410.

[80] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee, *Camera Distance-aware Top-down Approach for 3D Multi-person Pose Estimation from a Single RGB Image*, 2019, arXiv: 1907.11346v2, URL: https://github.com/mks0601/3DMPPE_POSENET_ (visited on 12/09/2020).

[81] Francesc Moreno-Noguer, « 3D Human Pose Estimation from a Single Image via Distance Matrix Regression », *in*: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1561–1570, DOI: 10.1109/CVPR.2017.170, arXiv: 1611.09010v1.

[82] L. Mourot et al., « A Survey on Deep Learning for Skeleton-Based Human Animation », *in*: *Computer Graphics Forum* 41.*1* (Feb. 2022), pp. 122–157, DOI: 10.1111/cgf.14426, arXiv: 2110.06901 [cs].

[83] Lucas Mourot et al., « JUMPS: Joints Upsampling Method for Pose Sequences », *in*: *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 1096–1103, DOI: 10.1109/ICPR48806.2021.9412160.

[84] Zaka-Ud-Din Muhammad, Zhangjin Huang, and Rashid Khan, « A Review of 3D Human Body Pose Estimation and Mesh Recovery », *in*: *Digital Signal Processing* 128 (Aug. 2022), p. 103628, DOI: 10.1016/j.dsp.2022.103628.

[85] Meinard Müller, Andreas Baak, and Hans-Peter Seidel, « Efficient and Robust Annotation of Motion Capture Data », *in*: *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '09: The ACM SIGGRAPH / Eurographics Symposium on Computer Animation, New Orleans Louisiana, Aug. 2009, pp. 17–26, DOI: 10.1145/1599470.1599473.

[86] Meinard Muller et al., *Documentation Mocap Database HDM05*, CG-2007-2, Universität Bonn, 2007.

[87] Lucie Naert, Caroline Larboulette, and Sylvie Gibet, « A survey on the animation of signing avatars: From sign representation to utterance synthesis », *in*: *Comput. Graph.* 92 (2020), pp. 76–98.

[88] Lucie Naert, Caroline Larboulette, and Sylvie Gibet, « LSF-ANIMAL: A Motion Capture Corpus in French Sign Language Designed for the Animation of Signing Avatars », *in*: *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2020, pp. 6008–6017.

[89] Lucie Naert, Caroline Larboulette, and Sylvie Gibet, « Motion Synthesis and Editing for the Generation of New Sign Language Content: Building New Signs with Phonological Recombination », *in*: *Machine Translation* 35.*3* (Sept. 2021), pp. 405–430, DOI: 10.1007/s10590-021-09268-y.

[90] Alejandro Newell, Kaiyu Yang, and Jia Deng, « Stacked Hourglass Networks for Human Pose Estimation », *in*: *Computer Vision – ECCV 2016*, ed. by Bastian Leibe et al., vol. 9912, Cham, 2016, pp. 483–499, DOI: 10.1007/978-3-319-46484-8_29.

[91] Thong Duy Nguyen and Milan Kresovic, « A Survey of Top-down Approaches for Human Pose Estimation », *in*: (Feb. 5, 2022), arXiv: 2202.02656 [cs], URL: http://arxiv.org/abs/2202.02656 (visited on 11/22/2023), preprint.

[92] Takuya Ohashi, Yosuke Ikegami, and Yoshihiko Nakamura, « Synergetic Reconstruction from 2D Pose and 3D Motion for Wide-Space Multi-Person Video Motion Capture in the Wild », *in*: *Image and Vision Computing* 104 (Dec. 2020), p. 104028, DOI: 10.1016/j.imavis.2020.104028.

[93] George Papandreou et al., « PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model », *in*: *Computer Vision – ECCV 2018*, ed. by Vittorio Ferrari et al., vol. 11218, Cham, 2018, pp. 282–299, DOI: 10.1007/978-3-030-01264-9_17.

[94] Georgios Pavlakos et al., « Expressive Body Capture: 3D Hands, Face, and Body From a Single Image », *in*: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, June 2019, pp. 10967–10977, DOI: 10.1109/CVPR.2019.01123.

[95] Dario Pavllo, David Grangier, and Michael Auli, « QuaterNet: A Quaternion-based Recurrent Model for Human Motion », *in*: *British Machine Vision Conference (BMVC)*, 2018, DOI: 10.48550/arXiv.1805.06485, arXiv: 1805.06485v2.

[96] Dario Pavllo et al., « 3D Human Pose Estimation in Video with Temporal Convolutions and Semi-Supervised Training », *in*: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, DOI: 10.1109/CVPR.2019.00794.

[97] Leonid Pishchulin et al., « DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation », *in*: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, June 2016, pp. 4929–4937, DOI: 10.1109/CVPR.2016.533.

[98] Davis Rempe et al., « HuMoR: 3D Human Motion Model for Robust Pose Estimation », *in*: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, Oct. 2021, pp. 11468–11479, DOI: 10.1109/ICCV48922.2021.01129.

[99] Nikolaos Sarafianos et al., « 3D Human Pose Estimation: A Review of the Literature and Analysis of Covariates », *in*: *Computer Vision and Image Understanding* 152 (Nov. 2016), pp. 1–20, DOI: 10.1016/j.cviu.2016.09.002.

[100] Saurabh Sharma et al., « Monocular 3D Human Pose Estimation by Generation and Ordinal Ranking », *in*: *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-Octob, 2019, pp. 2325–2334, DOI: 10.1109/ICCV.2019.00241, arXiv: 1904.01324.

[101] Xiaolong Shen et al., « Global-to-Local Modeling for Video-Based 3D Human Pose and Shape Estimation », *in*: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, June 2023, pp. 8887–8896, DOI: 10.1109/CVPR52729.2023.00858.

[102] Mingyi Shi et al., « MotioNet: 3D Human Motion Reconstruction from Monocular Video with Skeleton Consistency », *in*: *ACM Transactions on Graphics* 40.*1* (1 Feb. 28, 2021), pp. 1–15, DOI: 10.1145/3407659.

[103] Mingyi Shi et al., « PhaseMP: Robust 3D Pose Estimation via Phase-conditioned Human Motion Prior », *in*: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, Oct. 1, 2023, pp. 14679–14691, DOI: 10.1109/ICCV51070.2023.01353.

[104] Soshi Shimada et al., « Neural Monocular 3D Human Motion Capture with Physical Awareness », *in*: *ACM Transactions on Graphics* 40.*4* (4 Aug. 31, 2021), pp. 1–15, DOI: 10.1145/3450626.3459825.

[105] Soshi Shimada et al., « PhysCap: Physically Plausible Monocular 3D Motion Capture in Real Time », *in*: *ACM Transactions on Graphics* 39.*6* (6 Dec. 31, 2020), pp. 1–16, DOI: 10.1145/3414685.3417877.

[106] Leonid Sigal, Alexandru O. Balan, and Michael J. Black, « HumanEva: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion », *in*: *International Journal of Computer Vision* 87.*1-2* (Mar. 2010), pp. 4–27, DOI: 10.1007/s11263-009-0273-6.

[107] Jose Sosa and David Hogg, « Self-Supervised 3D Human Pose Estimation from a Single Image », *in*: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Vancouver, BC, Canada, June 2023, pp. 4788–4797, DOI: 10.1109/CVPRW59228.2023.00507.

[108] G. Stucki, T. Ewert, and A. Cieza, « Value and Application of the ICF in Rehabilitation Medicine », *in*: *Disability and Rehabilitation* 24.*17* (Jan. 2002), pp. 932–938, DOI: 10.1080/09638280210148594.

[109] G Stucki et al., « ICF-based Classification and Measurement of Functioning », *in*: *EUROPEAN JOURNAL OF PHYSICAL AND REHABILITATION MEDICINE* 44.*3* (2008).

[110] Ke Sun et al., « Deep High-Resolution Representation Learning for Human Pose Estimation », *in*: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, June 2019, pp. 5686–5696, DOI: 10.1109/CVPR.2019.00584.

[111] Xiao Sun et al., « Compositional Human Pose Regression », *in*: *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Oct. 2017, pp. 2621–2630, DOI: 10.1109/ICCV.2017.284.

[112] Xiao Sun et al., « Integral Human Pose Regression », *in*: *Computer Vision – ECCV 2018*, ed. by Vittorio Ferrari et al., vol. 11210, Cham, 2018, pp. 536–553, DOI: 10.1007/978-3-030-01231-1_33.

[113] Zhenhua Tang et al., « 3D Human Pose Estimation With Spatio-Temporal Criss-Cross Attention », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR, 2023, pp. 4790–4799.

[114] Mansour Tchenegnon, Sylvie Gibet, and Thibaut Le Naour, « A New Spatio-Temporal Loss Function for 3D Motion Reconstruction and Extended Temporal Metrics for Motion Evaluation », *in*: *European Conference on Computer Vision (ECCV 2022), Workshop on What Is Motion For?*, 2022.

[115] Bugra Tekin et al., « Direct Prediction of 3D Body Poses from Motion Compensated Sequences », *in*: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, June 2016, pp. 991–1000, DOI: 10.1109/CVPR.2016.113.

[116] Bugra Tekin et al., « Learning to Fuse 2D and 3D Image Cues for Monocular Body Pose Estimation », *in*: *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Oct. 2017, pp. 3961–3970, DOI: 10.1109/ICCV.2017.425.

[117] Alexander Toshev and Christian Szegedy, « DeepPose: Human Pose Estimation via Deep Neural Networks », *in*: *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, June 2014, pp. 1653–1660, DOI: 10.1109/CVPR.2014.214.

[118] The Ohio State University, *Advanced Computing Center for the Arts and Design Mocap Dataset*, https://accad.osu.edu/research/motion-lab/mocap-system-and-data.

[119] Gul Varol et al., « Learning from Synthetic Humans », *in*: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, July 2017, pp. 4627–4635, DOI: 10.1109/CVPR.2017.492.

[120] Ashish Vaswani et al., « Attention Is All You Need », *in*: *Advances in Neural Information Processing Systems* 2017-Decem.*Nips* (Nips 2017), pp. 5999–6009, arXiv: 1706.03762.

[121] Léon Victor, Alexandre Meyer, and Saida Bouakaz, « Learning-based pose edition for efficient and interactive design », *in*: *Computer Animation and Virtual Worlds* 32 (June 2021), DOI: 10.1002/cav.2013.

[122] Timo Von Marcard et al., « Recovering Accurate 3D Human Pose in the Wild Using IMUs and a Moving Camera », *in*: *Computer Vision – ECCV 2018*, ed. by Vittorio Ferrari et al., vol. 11214, Cham, 2018, pp. 614–631, DOI: 10.1007/978-3-030-01249-6_37.

[123] Bastian Wandt and Bodo Rosenhahn, « Repnet: Weakly Supervised Training of an Adversarial Reprojection Network for 3D Human Pose Estimation », *in*: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, 2019, DOI: 10.1109/CVPR.2019.00797, arXiv: 1902.09868.

[124] Jinbao Wang et al., « Deep 3D Human Pose Estimation: A Review », *in*: *Computer Vision and Image Understanding* 210 (Sept. 2021), p. 103225, DOI: 10.1016/j.cviu.2021.103225.

[125] Jingbo Wang et al., « Motion Guided 3D Pose Estimation from Videos », *in*: *Computer Vision – ECCV 2020*, ed. by Andrea Vedaldi et al., vol. 12358, Cham, 2020, pp. 764–780, URL: https://link.springer.com/10.1007/978-3-030-58601-0_45 (visited on 06/05/2023).

[126] Ruibin Wang, Xianghua Ying, and Bowei Xing, « Exploiting Temporal Correlations for 3D Human Pose Estimation », *in*: *IEEE Transactions on Multimedia* (2023), pp. 1–13, DOI: 10.1109/TMM.2023.3323874.

[127] Yong Wang et al., « Global and Local Spatio-Temporal Encoder for 3D Human Pose Estimation », *in*: *IEEE Transactions on Multimedia* (2023), pp. 1–11, DOI: 10.1109/TMM.2023.3321438.

[128] Ziming Wang et al., « Learning 3D Human Pose and Shape Estimation Using Uncertainty-Aware Body Part Segmentation », *in*: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, June 4, 2023, pp. 1–5, DOI: 10.1109/ICASSP49357.2023.10095635.

[129] Guodong Wei et al., « BoneNet: Real-time 3D Human Pose Estimation By Generating Multiple Hypotheses with Bone-map Representation », *in*: (2021), DOI: 10.14733/cadaps.2021.1448-1465.

[130] Xuan Wu et al., « LIDAR-based 3D Human Pose Estimation and Action Recognition for Medical Scenes », *in*: *IEEE Sensors Journal* (2024), pp. 1–1, DOI: 10.1109/JSEN.2024.3373192.

[131] Yuxin Wu et al., https://github.com/facebookresearch/detectron2, 2019.

[132] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh, « Monocular Total Capture: Posing Face, Body, and Hands in the Wild », *in*: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, June 2019, pp. 10957–10966, DOI: 10.1109/CVPR.2019.01122.

[133] Anastasios Yiannakides, Andreas Aristidou, and Yiorgos Chrysanthou, « Real-Time 3D Human Pose and Motion Reconstruction from Monocular RGB Videos », *in*: *Computer Animation and Virtual Worlds*, vol. 30, 3-4, May 1, 2019, DOI: 10.1002/cav.1887.

[134] Zhengdi Yu et al., « SignAvatars: A Large-scale 3D Sign Language Holistic Motion Dataset and Benchmark », *in*: *arXiv preprint arXiv:2310.20436* (2023).

[135] Jinlu Zhang et al., « MixSTE: Seq2seq Mixed Spatio-Temporal Encoder for 3D Human Pose Estimation in Video », *in*: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, June 2022, pp. 13222–13232, DOI: 10.1109/CVPR52688.2022.01288.

[136] Lijun Zhang et al., « Deep Semantic Graph Transformer for Multi-View 3D Human Pose Estimation », *in*: *Proceedings of the AAAI Conference on Artificial Intelligence* 38.7 (Mar. 24, 2024), pp. 7205–7214, DOI: 10.1609/aaai.v38i7.28549.

[137] Xuan Zhang et al., « Human 3D Pose Estimation Based on Sequence Graph Convolution », *in*: *2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, 2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, June 17, 2022, pp. 1034–1038, DOI: 10.1109/ITAIC54216.2022.9836467.

[138] Long Zhao et al., « Semantic Graph Convolutional Networks for 3D Human Pose Regression », *in*: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, June 1, 2019, pp. 3420–3430, DOI: 10.1109/CVPR.2019.00354, arXiv: 1904.03345.

[139] Qitao Zhao et al., « PoseFormerV2: Exploring Frequency Domain for Efficient and Robust 3D Human Pose Estimation », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 8877–8886.

[140] Ce Zheng et al., « 3D Human Pose Estimation with Spatial and Temporal Transformers », *in*: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, Oct. 2021, pp. 11636–11645, DOI: 10.1109/ICCV48922.2021.01145.

[141] Jieming Zhou et al., « Diff3DHPE: A Diffusion Model for 3D Human Pose Estimation », *in*: *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Paris, France, Oct. 2, 2023, pp. 2084–2094, DOI: 10.1109/ICCVW60793.2023.00223.

[142] Lu Zhou, Yingying Chen, and Jinqiao Wang, « Dual-Path Transformer for 3D Human Pose Estimation », *in*: *IEEE Transactions on Circuits and Systems for Video Technology* (2024), pp. 1–1, DOI: 10.1109/TCSVT.2023.3318557.

[143] Lu Zhou, Yingying Chen, and Jinqiao Wang, « SlowFastFormer for 3D Human Pose Estimation », *in*: *Computer Vision and Image Understanding* 243 (June 2024), p. 103992, DOI: 10.1016/j.cviu.2024.103992.

[144] Xiaowei Zhou et al., « Sparseness Meets Deepness: 3D Human Pose Estimation from Monocular Video », *in*: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, June 2016, pp. 4966–4975, DOI: 10.1109/CVPR.2016.537.

[145] Xingyi Zhou et al., « Towards 3D Human Pose Estimation in the Wild: A Weakly-Supervised Approach », *in*: *Proceedings of the IEEE International Conference on Computer Vision* 2017-Octob (2017), pp. 398–407, DOI: 10.1109/ICCV.2017.51, arXiv: 1704.02447.

[146] Yi Zhou et al., « On the Continuity of Rotation Representations in Neural Networks », *in*: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5738–5746, DOI: 10.1109/CVPR.2019.00589.

[147] Wentao Zhu et al., « MotionBERT: A Unified Perspective on Learning Human Motion Representations », *in*: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, Oct. 1, 2023, pp. 15039–15053, DOI: 10.1109/ICCV51070.2023.01385.

[148] Zhiming Zou and Wei Tang, « Modulated Graph Convolutional Network for 3D Human Pose Estimation », *in*: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, Oct. 2021, pp. 11457–11467, DOI: 10.1109/ICCV48922.2021.01128.

COLLEGE MATHSTIC
DOCTORAL BRETAGNE
BRETAGNE OCEANE

Université Bretagne Sud
ubs

Titre : Reconstruction de mouvements en 3D à l'aide de méthodes d'apprentissage profond : application aux handicaps moteurs

Mot clés : reconstruction de mouvement, apprentissage profond, handicap moteur, analyse et synthèse du mouvement

Résumé : Cette thèse porte sur la reconstruction de mouvement 3D à partir de vidéo RGB en utilisant des techniques d'apprentissage profond et propose une solution alternative aux systèmes capture de mouvement (MoCap) dans les situations complexes où leur utilisation est difficile, notamment le handicap moteur. La solution proposée est basée sur l'estimation de poses, suivie de la reconstruction du mouvement par un processus de correction qui préserve la continuité temporelle entre les poses successives. Nous définissons une fonction de perte Laplacienne pour entraîner les modèles d'IA, ainsi que des métriques d'évaluation de la temporalité du mouvement et la préservation de la structure du squelette. Le système de correction de mouvement développé en couplant modélisation Laplacienne du mouvement et apprentissage profond permet une reconstruction avec une meilleure qualité temporelle et spatiale. Dans un second temps, nous appliquons notre système de reconstruction de mouvement en situation de handicap moteur. Nous pallions le manque de données en générant des vidéos 2D à partir de mouvements 3D capturés par MoCap. Cette approche originale augmente nos données en utilisant plusieurs caméras d'enregistrement (différents points de vue) et des modèles 3D de personnages virtuels variés. Notre modèle de reconstruction montre de bonnes performances sur ces vidéos et offre des perspectives prometteuses pour des applications temps réel.

Title: 3D motion reconstruction with deep learning methods : application to motor disabilities

Keywords: motion reconstruction, deep learning, motor disability, motion analysis and synthesis

Abstract: This thesis deals with the reconstruction of 3D motion from RGB video using deep learning techniques and proposes an alternative solution to motion capture systems (MoCap) in situations where they are difficult to use. The proposed solution is designed based on human pose estimation, followed by motion reconstruction using a correction process that preserves temporal continuity between successive poses. We define a Laplacian loss function for training AI models, as well as metrics for assessing motion temporal features and skeletal structure preservation. The motion correction system developed by coupling Laplacian motion modeling and deep learning enables motion reconstruction with improved temporal and spatial quality. In a second step, we apply our motion reconstruction system to motor disability situations. We make up for the lack of data by generating 2D videos from 3D movements captured by MoCap. This original approach augments our data by using multiple recording cameras (different viewpoints) and various 3D models of virtual characters. Our reconstruction model performs well on these videos and offers promising prospects for real-time applications.