Data driven representation and synthesis of 3D human motion

Mathieu Marsot

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : MSTII - Mathématiques, Sciences et technologies de l'information, Informatique
Spécialité : Informatique
Unité de recherche : Laboratoire Jean Kuntzmann

Modèles statistiques pour representer et synthètiser des mouvement humains en 3D

Data driven representation and synthesis of 3D human motion

Présentée par :

Mathieu MARSOT

Direction de thèse :

Stefanie WUHRER — Directrice de thèse
CR, INRIA

Jean-Sébastien FRANCO — Co-directeur de thèse
Université Grenoble Alpes

Rapporteurs :

Florent DUPONT
PROFESSEUR DES UNIVERSITES, Université Lyon 1
Franck HETROY-WHEELER
PROFESSEUR DES UNIVERSITES, Université de Strasbourg

Thèse soutenue publiquement le **12 mai 2023**, devant le jury composé de :

Joelle THOLLOT PROFESSEUR DES UNIVERSITES, Grenoble INP	Présidente
Florent DUPONT PROFESSEUR DES UNIVERSITES, Université Lyon 1	Rapporteur
Franck HETROY-WHEELER PROFESSEUR DES UNIVERSITES, Université de Strasbourg	Rapporteur
Silvia ZUFFI DOCTEUR EN SCIENCES, Istituto di Matematica Applicata e Tecnologie Informatiche (IMATI-CNR)	Examinatrice
Anne-Hélène OLIVIER MAITRE DE CONFERENCES, Université Rennes 2	Examinatrice

Invités :

Jean-Sebastien FRANCO
MAITRE DE CONFERENCES, INRIA
Stefanie WUHRER
CHARGE DE RECHERCHE, INRIA

Abstract

In this manuscript, we propose data-driven methods to generate realistic 3D human motion. Although data-driven approaches are difficult to implement without sufficient data, the rapid increase in 3D human data makes them increasingly attractive. While generative modelling of dense human bodies has been widely studied over the past decades, it has become apparent that representing motion as a sequence of static poses has limitations when processing inputs with missing images, occlusions or low spatial resolution. Regarding motion, existing work has mainly focused on sparse representations of the human body, such as a set of sparse surface markers or a skeleton. We investigate the representation and synthesis of spatially dense human motion. We specifically target bodies wearing tight clothing and focus on modelling dense human motion as the temporal evolution of a sparse body pose and its interaction with a dense body morphology. To do so, we exploit the power of neural representations, which have allowed for impressive progress in 2D image processing and have shown promising first results on 3D data.

Modelling motion from data poses many challenges. Firstly, spatio-temporal motion data is high dimensional. Second, human motion and morphological variability are interdependent. And third, the formalism of what makes a motion realistic is not well known, which complicates the translation of this problem into optimization objectives.

To address these issues, we propose three contributions. The first two investigate the use of low-dimensional latent spaces to represent and synthesize human motion. Specifically, in our first contribution, we propose to encode a temporal segment of motion into a single latent vector while applying a Gaussian assumption on the distribution of the data in the latent space. This generic latent representation allows the generation of new motions and also allows the completion and denoising of sparse data.

In the second contribution, we extend the first work to longer motions representing them as a sequence of latent vectors, where each vector characterizes a temporal segment of the movement. This approach greatly improves the accuracy of the model and the results on the completion task.

In a third and final contribution, we study motion transfer, where the goal is to automatically transfer the motion of a source character to a target character. In this approach, we propose an alternative to the existing approaches which directly operate on the dense surface by operating both on the dense surface and an intermediate skeletal representation.

The source code implementing the different reconstruction methods is released as open source software for research purposes.

Résumé

Dans ce manuscrit, nous proposons des méthodes basées sur les données pour générer des mouvements humains réalistes en 3D. Bien que les approches basées sur les données soient difficiles à mettre en œuvre en l'absence de données suffisantes, l'augmentation rapide des données humaines en 3D les rend de plus en plus attrayantes. Bien que la modélisation générative des corps humains denses ait été largement étudiée au cours des dernières décennies, il est devenu évident que la représentation du mouvement comme une séquence de poses statiques présente des limites lors du traitement d'entrées avec des images manquantes, des occlusions ou une faible résolution spatiale. En ce qui concerne le mouvement, les travaux existants se sont principalement concentrés sur des représentations éparses du corps humain, telles qu'un ensemble de marqueurs de surface épars ou un squelette. Nous étudions la représentation et la synthèse des mouvements humains spatialement denses. Nous ciblons spécifiquement les corps portant des vêtements serrés et nous nous concentrons sur la modélisation du mouvement humain dense comme l'évolution temporelle d'une pose corporelle peu dense et son interaction avec une morphologie corporelle dense. Pour ce faire, nous exploitons la puissance des représentations neuronales, qui ont permis des progrès impressionnants dans le traitement des images 2D et ont donné des premiers résultats prometteurs sur les données 3D.

La modélisation du mouvement à partir de données pose de nombreux défis. Tout d'abord, les données de mouvement spatio-temporelles sont de haute dimension. Deuxièmement, le mouvement humain et la variabilité morphologique sont interdépendants. Et troisièmement, le formalisme de ce qui rend un mouvement réaliste n'est pas bien connu, ce qui complique la traduction de ce problème en objectifs d'optimisation.

Pour répondre à ces questions, nous proposons trois contributions. Les deux premières portent sur l'utilisation d'espaces latents de faible dimension pour représenter et synthétiser le mouvement humain. Plus précisément, dans notre première contribution, nous proposons d'encoder un segment temporel de mouvement dans un seul vecteur latent tout en appliquant une hypothèse gaussienne sur la distribution des données dans l'espace latent. Cette représentation latente générique permet de générer de nouveaux mouvements ainsi que de compléter et de débruiter des données éparses.

Dans la deuxième contribution, nous étendons le premier travail à des mouvements plus longs en les représentant comme une séquence de vecteurs latents, où chaque vecteur caractérise un segment temporel du mouvement. Cette approche améliore considérablement la précision du modèle et les résultats sur la tâche de complétion de données.

Dans une troisième et dernière contribution, nous étudions le transfert de mouvement, où le but est de transférer automatiquement le mouvement d'un personnage source à un personnage cible. Dans cette approche, nous proposons une alternative aux approches existantes qui opèrent directement sur la surface dense en opérant à la fois sur la surface dense et sur une représentation intermédiaire du squelette.

Le code source mettant en œuvre les différentes méthodes de reconstruction est publié en tant que logiciel libre à des fins de recherche.

”Dédicace”

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Representation and synthesis of 3D humans in motion is a long-standing problem of 3D computer vision. 3D human motion is a vast subject and in this manuscript, we investigate data driven models of motion that model the outer surface of 3D human bodies often referred to as body geometry. We are neither interested in modelling the underlying anatomy nor the appearance of the body but focus on the evolution of its geometry in space and time. More specifically, we propose generative representations of the human that can infer the dense body geometry from sparse acquisitions or that automatically generate realistic body motions. When considering 3D human modelling, there are currently two major approaches:

- Physics based approaches which leverage biomechanical constraints and physical simulations,
- Data driven approaches which focus on statistical modelling of human body geometry.

In this work, we focus on data-driven approaches. The data driven methods are statistical models of the body surface and its deformations and are usually faster than the physics-based approaches [3, 4]. The downside of data-driven models is that they require a significant amount of 3D motion data to generalize. With 3D data becoming more and more accessible due to the release of huge 3D human motion datasets [5, 6], these methods have gained in popularity in the recent years.

Such models are useful to many industries. In the video game industry, most games now allow the player to evolve in 3D scenes filled with human characters. Generating plausible 3D characters in motion to populate these scenes is time-consuming, and the generated characters are often predetermined and may lack variability. Automatic synthesis of plausible motions is an attractive solution to accelerate the design process and introduce a natural variability in the generated motions. In the movie industry and digital entertainment industries, a common

Figure 1.1: Left : picture of the Kinovis multi-view platform. Right : Acquisition pipeline, an actor is filmed under multiple angles by 68 RGB cameras. A 3D point cloud is then computed from the images using the 3D reconstruction algorithm proposed in [1].

practice is to animate digital fictional characters using the motion of a performing actor which is transferred to a virtual avatar. This requires to extract the motion information of a character independently of its real body shape, and often requires a lot of manual input. Using structured representation of human motion, it is possible to automate this process and facilitate the work of 3D artists. Other potential applications that could benefit from the representations proposed in this thesis can be found in the medical field, where out of distribution motion detection can help for automatic diagnosis of some diseases, or for automatic try-on where clothes could be virtually added on top of animated customers 3D avatars to estimate the best clothes mensuration and limit clothes returns.

1.2 Motion Data

At the core of any statistical model lies the data. In this manuscript, we will focus on the temporal evolution of the body surface. This surface is a 2D manifold embedded in 3D space which deforms in time. However, 3D motion acquisition setups (e.g. LIDAR, multi-view platforms) provide a discretized capture of this manifold. Their output is often a sequence of 3D point clouds which is a 3D spatio-temporal sampling of the deforming body surface.

Motion acquisitions systems are often categorized into two categories : Active systems and Passive systems. Passive systems only capture the environment light while active systems require additional light sources (*e.g.*infrared, colored lighting). While active systems tend to allow for more accurate capture, passive systems are easier to set up and scale better. In this thesis, we focus on processing the output of passive multi-view acquisition systems. Multi-view systems estimate the moving surface by triangulating points from videos acquired by a set of carefully calibrated RGB cameras which film a moving actor. We refer to the task of extracting a sequence of point cloud from these videos as 3D reconstruction [1, 7, 8] and give an overview of the pipeline of our multi-view acquisition platform (the Kinovis platform[1]) in Figure 1.1.

The point cloud output by these multi-view systems are spatially unaligned. Here, spatially aligning a motion sequence consists in providing anatomically

[1] http://kinovis.inria.fr/inria-platform/

corresponding points the same label (e.g. by the index in the ordering of a triangle mesh). When considering motion, it means providing correspondences between bodies from the same motion sequence or more generally between bodies of various actors performing various motion sequences. In the following, we often refer to unaligned data as correspondence-free data or unstructured data.

To train data-driven methods, it is easier to work with spatially aligned data. One way of putting 3D point clouds in correspondence is to fit a template based body model [9,10] to the data by deforming a template that best explain the observed data. Figure 1.2 shows an example of body model fitting on a walking sequence. Model fitting on correspondence free data is prone to local minima but can be improved using priors of human motion (cf. Chapter 4,5).

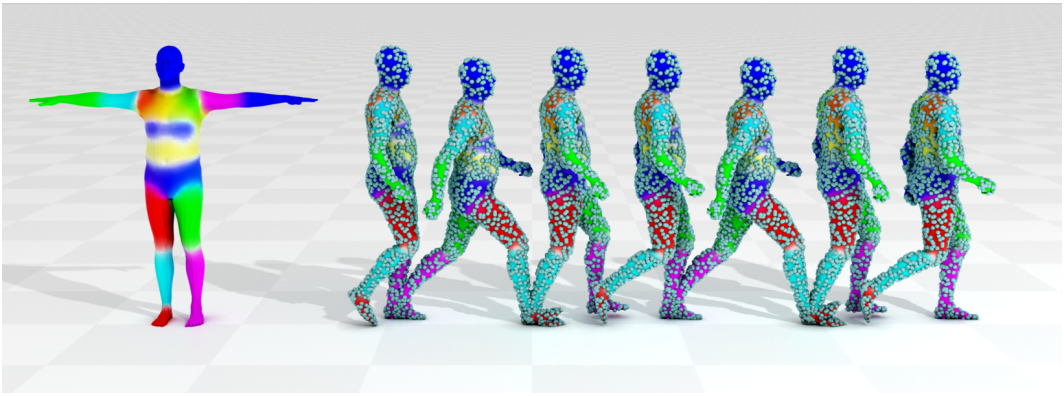


Figure 1.2: Spatial alignment of a walking sequence using template mesh fitting. The template mesh (left) is fitted to a sequence of unaligned point clouds. The resulting mesh sequence provides correspondences between the bodies as illustrated by the shared color-coding.

1.3 Problem statements

In this manuscript, we will address two major problems of 3D motion modelling. The first problem is related to spatio-temporal alignment of motion data and is referred to as **Motion completion**. The second problem is related to automatic **Motion generation** which we approach by different techniques like motion interpolation, motion sampling and motion retargeting.

Motion completion This first problem we tackle is illustrated in Figure 1.2. The objective is to complete a sparse spatio-temporal sampling of a human motion by fitting a statistical body model to the observed point cloud data. This problem is especially challenging for motions with complex pose and orientation variations as the fitting is prone to local minima which are difficult to disambiguate using a static per frame fitting. We show through the chapters of this manuscript that leveraging the temporal context of motion is beneficial in this task.

To benchmark our methods on this task, we leverage real data that we acquired using the Kinovis multi-view acquisition platform. We evaluate the robustness

of our representations against various spatio-temporal sparsity by decreasing the acquisition frame rate and point cloud density.

Motion generation A second problem we investigate is the automated generation of human motion. Current 3D animation often represent human bodies using a 3D mesh template which is then animated manually. In our work, we investigate two generative approaches : automatic generation of a character animation by leveraging structured latent representations of motions and motion retargeting which aims at generating new motions by transferring the existing motion of a source character to a target character.

1.4 Challenges

Statistical modelling of 3D humans in motion is a complex task with different intertwined sources of variability. The variability is often divided in two major factors. The body shape, characterizing the morphology and clothing of the person and the evolution of body pose, called motion in the following, which characterizes the underlying actions that are performed by the person. In this work, we are not interested in dynamic details like soft-tissue deformation. Another way to present it is that we assume the body shape of the person to remain constant through the motion. Even under this assumption, modelling the overall evolution of the human body presents many challenges. A major challenge is to disentangle the motion and morphology variability which are tightly intertwined [11]. More related to the spatio-temporal nature of human motion, a challenge is the 4D reasoning which drastically increases the computational complexity over the analysis of static 3D poses, especially when considering long and various motions. This raises questions such as what is the best way to segment a motion spatio-temporally? More related to the generative tasks, another difficulty is to characterize what a realistic motion is. While perceptual studies of human motion are a long-standing research problem [12], the mathematical formalism of what makes a human motion realistic is not well-defined. Realism therefore still is a relatively subjective assessment which is hard to translate in terms of optimization objectives.

1.5 Contributions

To address these challenges, we propose three contributions which are listed below.

A structured latent space for human body motion generation In this first contribution presented in Chapter 4, we propose a structured latent representation for motion generation which allows projecting sequences of meshes into a latent representation that disentangles the influence of shape and morphology. This first work achieves state-of-the-art results for the motion completion task from spatio-temporally sparse inputs, allowing to recover the dense geometry of the body from sparse spatio-temporal point clouds, and the structure of the latent space also allows for natural motion generation by linear interpolation in latent space. This work was published at the International Conference on 3D Computer Vision 2022 [13].

Representing motion as a sequence of latent primitives The model proposed in our first contribution does not handle long motions and only considers variety of motions that exhibit a cyclic hip motion. In a second contribution presented in Chapter 5, we propose to include more motion variety and to handle longer duration by factoring the temporal complexity using a sequential latent representation that automatically represents a motion as a sequence of latent primitives. This work performs well on the motion completion task for any kind of motion, allowing to automatically recover the spatially aligned dense geometry of long sequences for any complex motions. This work is currently under review at the International Conference on Computer Vision 2023 [14] .

Correspondence-free online human motion retargeting In the third contribution presented in Chapter 6, we investigate the more specific problem of motion retargeting which aims to realistically animate a target character with the motion of a source character. While the first two methods propose a latent representation of motion segments, this last work operates per frame, retaining temporal context about the motion. This last work directly operates on sequences of point clouds without requiring correspondence information between different frames of the motion and showed outperforms the state-of-the-art when applied to unseen data acquired with the Kinovis multi-view platform. That contribution was achieved in close collaboration with fellow PhD student Rim Rekik Dit Nekhili and is currently under review at the International Conference on Computer Vision 2023 [15].

2
Related works

Modelling 3D Human motion is a complex problem. To approach it, the first step is typically to choose a suitable human body representation and an animation model that can deform the body. Then, it is possible to automatically synthesize plausible body animations using various statistical models. The first section of this chapter will focus on the representation and synthesis of human bodies.

Armed with a body representation and animation model, it is then possible to model the evolution of the body through time hence modelling human motion. There are multiple strategies to model human motion either sparsely or densely in space depending on the desired applications. We will detail the different motion modelling strategies in the second section.

2.1 Human body modelling

The literature of 3D body modelling is vast and can roughly be divided into the modelling of three major factors.

- Their geometry, which corresponds to the 2D surface of a 3D volume and is often referred to as a skin or body surface
- Their appearance which is related to the outside color
- Their inner composition like bone structure and inner tissues

These three modalities are intertwined. For instance, the 3 following works [7, 16, 17] try to deduct geometry from the appearance from one or multiple RGB images. In [18], Keller *et.al* shows that it is possible to infer a plausible structure of the inner bone structure solely from the body geometry. In this manuscript, we focus on modelling the evolution of the body surface through time, so the discussion below will focus on popular body representations and generative models that operate on the body geometry.

2.1.1 Representation and animation of 3D bodies

Meshes and skinning A first family of methods parameterize the body geometry with a 3D mesh. The most common animation process for body meshes transfers the deformation of an underlying skeleton to the vertices of the body mesh. The underlying skeleton is characterized by a set of 3D joints and a joint hierarchy. The skeleton joints are linked to the mesh with skinning weights [19, 20] which associate each vertex to the joints. The skeleton is then animated by an artist or a motion model and the skeleton animation is transferred to the mesh using blending techniques [21–23]. Blending techniques transfer the averaged local rigid transformation of the corresponding joints to the associated vertices, for the result to be realistic, this animation process requires a careful design of the mesh, skeleton and skinning weights.

As designing mesh templates and rigging them to a skeleton is a time-consuming task, some methods proposed to automate the rigging task by automatically predicting the skinning weights given a body mesh and its skeleton. Some works used constraints based on the distance between joints and vertices [24, 25] to estimate the weights. More recently, Xu *et.al* [26] went even further, predicting both the skeleton and the skinning weights using neural networks.

Mesh based animation is practical because it can easily be incorporated and rendered by current graphic pipelines. However, with the growing accessibility of 3D motion datasets [5, 6], machine learning approaches have grown more and more appealing. Currently, meshes are however still hard to process with current gradient-descent based models as they lack the spatial structure of images and their varying topology makes them difficult to process. Some interesting neural layers were proposed to handle mesh representation by exploiting graph theory e.g. [27,28] or generalizing convolutions to graph structures e.g. [29, 30]. To better process the 3D data with recent data-driven models, there was a resurgence of implicit shape representations.

Implicit representations In implicit representations, the 3D shape is implicitly characterized as a level set of a scalar field of the 3D space. Recently, neural networks have been used to approximate these fields. Implicit fields cover the full 3D space as they assign a scalar value to each position in 3D space. They can indicate distance information from the surface e.g. [31] or occupancy information when considering watertight surfaces e.g. [16, 32]. The advantage over meshes is that this representation can represent any topology and does not rely on a specific mesh template. Neural fields are however not easily incorporated in existing graphics pipelines and in the case of human body modelling, neural fields are hard to animate. To incorporate them in graphics pipelines, it is possible to extract a mesh from a spatial sampling of the learned implicit function using the marching cubes algorithm [33]. Then to solve the animation problem, Deng *et.al* [34] proposed to specialize a neural field to human bodies, representing a body as an articulated implicit shape which can be animated using a kinematic skeleton.

Deformation transfer While skeleton based animation is popular, another line of work animates bodies by transferring the deformation from a source body to a target body by directly operating at a surface level. In this case, the deformation

information is often a mix of data-driven and prior assumptions such as near-isometric deformations of a fixed person. Deformation transfer can then be done either by optimizing mesh deformations [35–40] or by using representations of 3D body shape that disentangle shape and pose [41, 42].

The first way to optimize deformations [35–38] is to consider motion retargeting as pose deformation transfer, with the objective of transferring the pose related deformation of the source character to the target character while preserving its identity. Other works [39,40] consider motion retargeting as an identity deformation transfer, with the objective of transferring the identity of the target body to the source body.

2.1.2 Generating 3D bodies

As discussed above, we consider two ways of representing human bodies, the first is using a mesh parameterization, the second using neural fields. With either representation, the task of automatically generating body meshes present two main challenges :

- Generation of a plausible body morphology (or body shape), which characterizes the identity of a person
- Generation of a realistic body pose, which characterize the shape agnostic deformation.

To generate plausible body shapes, first data-driven representations of the identity of a 3D body performed principal component analysis on a dataset of aligned meshes, leading to a compact representation with a generative shape space that proved useful in a variety of applications [43]. The success of this initial body model led to the inclusion of pose variation *e.g.* [9, 10, 44, 45]. Of special interest to our work, is the SMPL body model [10] which parameterized the pose variation with a kinematic skeleton making it easy to integrate in the existing mesh animation software.

Extensions of these generative body models were proposed, sometimes even encoding hand pose and facial expressions *e.g.* [46,47] and soft tissue deformations [48,49]. Given as input marker data from various motion capture datasets, [6] fitted various body models to a vast collection of bodies provides the community access to a large human motion dataset. Other recent works in body modelling leverage deep learning techniques *e.g.* [41, 50, 51], and can decouple variations due to body pose, shape, and soft tissue deformations.

Among the previous body models, [41, 46, 51] include a generative prior of pose parameters in addition of the generative shape spaces. This allows not only to generate new body identities but to automatically generate realistic body poses as well. In these works, the pose latent space follows a Gaussian prior similarly to the variational autoencoder (VAE) introduced in [52]. A follow-up work on pose priors [53] proposed to use generative adversarial networks (GAN) [54]. Unlike VAEs, GANs use an implicit probability distribution of the latent variables allowing to relax the Gaussian assumption on the prior distribution. In an orthogonal approach, [55] proposed a non-probabilistic model which directly represents the

manifold of plausible poses using a neural field which characterizes plausible poses in a latent representation, leading to an even more expressive prior.

While recent mesh based generative models can generate detailed human bodies, they are restricted by the fixed topology of meshes, limiting their generalization capabilities. [56] and [57] therefore proposed implicit priors relying on neural fields that are not limited by the fixed topology.

2.2 Human Motion Modelling

Given a body representation and deformation model, the straightforward approach to represent human motion is to represent it as a sequence of independent poses of a body template, where each body pose does not exploit information about its previous poses. Doing so, the lack of temporal context tends to be problematic. For instance, for the task of mesh estimation from images, a model which provides a body mesh for each image is prone to artifacts like temporal jitter and can hardly disambiguate occlusions. To solve this, efforts were made to model the evolution of bodies through time and incorporate a temporal dimension to the body models, transforming them into motion models.

The research on human motion can roughly be divided into two lines of work. *Spatially sparse* motion modelling which encompasses methods that learn the structure of human motion on a representation that is sparse in 3D space, like a kinematic skeleton or sparse body surface markers. This is especially useful in tasks where the focus is on the evolution of body pose, such as action recognition or pose estimation. It however discards the geometric details unlike *Spatially dense* motion modelling. These methods model the full evolution of the dense 3D body. Modelling the evolution of the dense surface makes it harder to disentangle pose related variability from shape related variability and drastically increases the required computational power. These methods are thus better suited when considering short term motions where body shape has to be precisely captured.

In our contributions, we stand in between dense and sparse modelling. We assume body shape constant through the motion, which brings us close to sparse modelling. But we retain the correlation between the body shape and the body pose and train with objectives that accounts for the dense body geometry, bringing us close to spatially dense modelling.

2.2.1 Sparse motion modelling

There is a variety of models that focus on generative modelling of sparse motion. One line of work proposes to model high level motion patterns and allows to directly generate motion by reproducing these patterns. Another line of work proposes to automatically transfer the motion of a skeletal representation to another skeletal representation. Finally, a last line of work proposes to build generic motion priors which encode skeletal motion into a structured latent space.

Direct motion generation Spatially sparse human motion models proposed different data-driven methods to synthesize motion patterns of skeletal representations or sparse marker positions *e.g.* [11,58–61]. Leveraging the structure

and low dimensionality of these sparse representations, these works effectively learn the structure of human motion over durations of multiple seconds allowing the model to have a high level understanding of the performed motion. [62] proposed an action based conditioning that considers a set of labeled actions to learn motion generation based on action labels, which was further extended [63] to condition motion generation based on textual inputs by leveraging the progress of natural language processing.

Motion priors Sparse motion priors are a family of methods that encode temporal sequences of pose parameters of static body models into a low dimensional representation. Focusing on small temporal context, [64] and [65] proposed priors that encode two frames of motion into a latent representation. To give a better intuition about their work, [64] proposes to interpret this prior as a transition prior between two consecutive poses. Most similar to our first two contributions are methods that build motion priors over multiple seconds of 4D human motion data. [66–69]. [66, 67] consider motions of a fixed duration and encode them in a motion space, which captures information about pose changes over time. [68] is restricted to 4 different motions but proposes to consider longer motion by representing them as a sequence of motion words with fixed duration. [69] proposes a temporally continuous representation to allow for virtually infinite temporal resolution of the generated motions. These priors are at the frontier between fully dense methods and sparse methods as they consider the dense geometry at training time, but do not leverage it at inference.

Motion transfer In a deformation transfer approach, based on skeleton, Gleicher *et.al* [70] proposed to transfer the skeletal motion from one person to another by considering an optimization problem with kinematic constraints over the entire motion sequence. Follow-up works [71–74] improved the optimization and introduced hand-designed kinematic constraints for particular motions. With the surge accessibility of captured motion data and the efficiency of deep learning techniques, latest data-driven approaches [75–78] showed outstanding results without requiring handcrafted energies. Some of these data-driven approaches [75, 76] require paired training data, whose design involves human effort. Therefore, another line of works [77, 78] proposed unsupervised transfer strategies. Villegas *et.al* [77] propose an unsupervised framework based on adversarial cycle consistency to ensure plausibility of the motions. This method generates natural motions for unseen characters, but only operates at a skeletal level and does not include geometric details.

2.2.2 Dense Motion Modelling

Direct motion generation Over the past few years, a number of works proposed studying 4D human motion data that is densely sampled in space. The first work to tackle this problem [79] combines two linear models: one capturing dense static 3D shape data and one capturing the motion of MoCap markers. The two linear models are coupled based on semantic parameters including weight and height, which allows generating dense 4D human motion sequences. With dense 4D data

becoming increasingly available in recent years, a number of studies propose data-driven methods trained on dense data. First methods including [80–83] train on either a single motion sequence or multiple sequences showing the same subject performing different motions. A recent work that studies motions of a single subject proposes a deep latent variable model for 4D human motion synthesis [68] to model the probabilistic character of motion.

Motion priors To incorporate dense surface information in motion prior, one line of work uses implicitly defined surfaces over time to learn from raw 4D sequences [84–86], and successfully process human motion data. However, the high dimensionality of the 4D data constrains the sequences to a few frames.

The second line of work that we investigate is a variation of the sparse motion priors. In our first contribution [13], we investigate learning a motion space for 4D sequences of varying duration which combines a sparse prior on the pose parameters and the shape space of a static body model to retain the dense information of the motion. We demonstrate experimentally that this motion space outperforms similar motion priors [66] and [67] for the task of motion completion and allows for realistic motion transfer and interpolation. Our second contribution is a sequential prior continuous in time that shares the intuitions of [68] for sequential modelling of motion and [69] for the benefits of a temporally continuous representation. This second work focuses more on motion completion, and we show that by leveraging our prior, we can recover complex motions of up to 5 seconds outperforming existing methods on correspondence-less data.

Motion transfer Improving upon the sparse motion transfer, [87] proposes to include geometry and investigates hybrid skeleton-based motion retargeting with both a data-driven network and a post inference optimization based on the dense geometry to preserve self-contacts and prevent interpenetration. Improving on the static dense shape transfer approach, recent works [88, 89] add temporal context by considering $3 - 4$ consecutive frames when transferring the character identity. These motion transfer approaches however require the source motion to be in correspondence to leverage the temporal context. Our third contribution proposes an alternative motion transfer approach which has the advantage of directly operating on correspondence-less data and considering higher level motion features by considering 30 consecutive frames of motion. We show that incorporating this longer context leads to better motion transfer and that the correspondence-less approach allows us to operate directly on the raw output of multi-view platforms.

3 Background

This section provides technical background used in the different methods of this thesis. First we will present the SMPL body model [10], a widely used parametric human body model which we leverage as body representation in Chapter 4 and 5. Then we provide an high-level presentation of various neural network architectures that we leverage in our three contributions.

3.1 SMPL body model

Our representations build upon the SMPL body model. This model parameterizes the human geometry using a mean 3D template shape T in a T-Pose whose deformations are controlled by two independent parameters: shape parameters β and pose parameters θ.

3.1.1 Training data

The model parameters were learnt from two datasets : the CAESAR dataset [90] which is referred to as a "multi- shape" dataset, and a "multi-pose" dataset. The used examples from the "multi- shape" dataset amount to a total of 1700 male body geometries and 2100 female body geometries. All the shapes from this dataset are in a standing A-pose with the arms slightly offset from the chest. The "multi-pose" dataset consist of 40 different people performing various body poses for a total of 1786 body geometries (891 female bodies, and 895 male bodies). All the body geometries have been spatially aligned with the mean template T using the registration method of Bogo *et.al* [91].

Using these datasets, 3 different versions of SMPL were learnt. Two with gender specific templates (Male, Female) and a universal one (Neutral). In this manuscript, we only consider the Neutral SMPL model.

3.1.2 Shape parameters β

The shape parameters control the morphology of the body and are a low dimensional PCA representation of the body morphology variations in T-Pose learnt from the "multi-shape" dataset.

Given the mean template T and shape parameters β, it is possible to generate a shaped template in T-Pose $\hat{T}$ using the shape parameters such as : $\hat{T}(\beta) = T + B_S(\beta)$ with B_S the linear function that corresponds to the inverse PCA transformation matrix.

This latent shape representation was shown to be semantic to some extent. For instance, the first component of the Neutral model mostly controls height and gender while its second component mostly affects body weight. This representation can also be leveraged to generate unseen shape by using linear interpolation in the latent space or by sampling latent representations.

3.1.3 Pose parameters θ

To repose the shaped template, SMPL relies on a kinematic skeleton and linear blend skinning (LBS) [92]. The pose parameters θ control the 3D rotations that characterize the pose of the kinematic skeleton. More specifically, given a shaped template, a set of 3D joints are extracted from the template vertices using a learnt joint regressor function $\mathcal{J}(\beta)$ which takes the shape parameters β as input and outputs the 3D joint positions in T-Pose. Then, using linear blend skinning and a learnt skinning matrix $\mathcal{W}$, the skeleton deformations are transferred to the shaped template generating a posed body mesh $M(\theta, \beta) = LBS(\hat{T}(\beta), \mathcal{J}(\beta), \theta, \mathcal{W})$.

The joint regressor function and skinning matrix are both learnt using the multi-pose dataset.

3.1.4 Pose-Corrective blendshapes

Linear blend skinning is known to cause artifacts around the joints due to the linear blending of rotations. To tackle these artifacts, some works used different rotation representation for the joint rotations θ (eg. log-matrices [93], dual-quaternions [21]). In SMPL, the authors instead propose to use pose corrective blendshapes which are corrective offsets learnt as a linear function [94] of the pose parameters θ expressed in the axis-angle representation of rotations. By noting $B_P(\theta)$ the pose corrective blendshape function, the full equation of the SMPL model is :

$$M(\theta, \beta) = LBS(\hat{T}(\beta) + B_P(\theta), \mathcal{J}(\beta), \theta, \mathcal{W})$$

.

3.1.5 Body generation with SMPL

Using the SMPL body model, it is therefore possible to generate a dense 3D human body geometry using a low dimensional parameterization β, θ. In Chapters 4 and 5, we leverage this low dimensional body parameterization to reduce the computational cost of processing high dimensional 3D body sequences. Another major advantage of this generation process is that it is differentiable *w.r.t* β and

θ, making it easy to integrate in learning methods which rely on gradient descent optimization.

3.2 Variational Autoencoders

In Chapters 4 and 5, we propose to leverage Variational Autoencoder (VAE) architecture [52] to learn latent representations of motion. VAEs are a generative variant of the Autoencoder (AE) model [95].

Autoencoders are self-supervised neural networks used to learn a low dimensional latent representation of a dataset. Usually, they are divided into two neural functions. An encoder function E that allows to convert a data point x_i to a latent representation z_i and a decoder function D that converts a latent vector to a data point such as : $z_i = E(x_i), x_i = D(z_i)$.

To train an AE, the objective is to approximate D and E using neural networks which minimizes the loss $\mathcal{L}_{AE} = \sum_i |D(E(x_i)) - x_i|$.

AEs can be leveraged as generative models by using their latent representation to generate a new data point $\hat{x}$ using the decoder function D and a sampled latent representation z. However, the generated point $\hat{x}$ is not guaranteed to fall into the distribution of realistic data points. To improve the realism of generated data points, Variational autoencoders add a constraint on the latent prior distribution $p(z)$ of realistic data points. Using this latent prior, it is then possible to sample realistic data points by sampling $p(z)$.

To encourage this constraint, the encoder network output is interpreted as a probability distribution which tries to approximate the true posterior probability distribution $p(z|x_i)$. In our case, this posterior distribution is intractable as we wish to approximate the likelihood $p(x|z)$ by a non linear neural network so the expectation maximization algorithm [96] cannot be used.

Therefore, the posterior distribution is approximated by a probabilistic encoder network such as as $E(x_i) = q(z|x_i)$. To encourage the encoder to approximate the true posterior distribution the objective is to minimize $KL(q(z|x_i), p(z|x_i))$, with KL the Küllback-Leibler divergence. As the true posterior is unknown, this term cannot be optimized directly but it can be shown that a variational lower bound ($ELBO$) of this term is:

$$ELBO = -KL(q(z|x_i), p(z)) + \mathbb{E}(log(p(x_i|z))$$

The objective is then to maximize this lower bound to match the approximated posterior distribution and the true posterior as much as possible. In an autoencoder parlance, the expectancy term can be interpreted as a negative reconstruction loss such as maximizing the $ELBO$ is equivalent to minimizing the loss :

$$\mathcal{L}_{VAE} = \sum_i KL(q(z|x_i), p(z)) + \mathcal{L}_{AE}.$$

Using this formalism, it is common to make a Gaussian assumption on the probability distribution of the posterior $q(z|x_i)$ and the latent prior $p(z)$ to obtain an analytical form for the KL divergence which can be easily optimized in a gradient descent optimization. The common assumption introduced in [52] is to parameterize $q(z|x_i)$ by a a multivariate Gaussian with a diagonal covariance

structure $q(z|x_i) = \mathcal{N}(\mu(x_i), \sigma(x_i))$ and the latent prior by an isotropic centered gaussian $p(z) = \mathcal{N}(0, I)$.

The VAE formalism introduces a trade-off between the reconstruction quality and the realism of generated data points. A fixed weight w_{KL} is often used to control this trade-off [97] and hence the VAE loss becomes :

$$\mathcal{L}_{VAE} = w_{KL} \sum_i KL(q(z|x_i), p(z)) + \mathcal{L}_{AE}.$$

3.3 Sequence to sequence architectures

In the following Chapters, our input human motion data is always given as a temporal sequence of bodies. Processing sequential data using machine learning is challenging because different input motions exhibit different numbers of frames introducing variability in the input dimension.

In Chapter 4, we propose a temporal alignment that converts each motion sequence to the same number of frames. By doing so, it is possible to leverage multi-layer perceptrons that require a fixed input dimension but it restricts the network to a specific temporal alignment limiting the variety of motion it can consider. In Chapter 5 and 6, to loosen the alignment constraint, we leverage two different sequence to sequence architectures. Sequence to sequence architectures are a category of network architectures that can handle variable length sequences without assuming independence of the sequence elements.

3.3.1 Recurrent networks

In Chapter 6, we use a recurrent neural network (RNN) to transfer the skeletal motion of a source character to a target character. Recurrent networks have been studied since the nineties [98] and are a family of architecture that retain context information about the sequence.

In most RNNs, an input sequence $\{x_i\}_{i=0}^N$ is input iteratively where each element x_i is associated to an hidden state h_i that retains information about the predecessor elements of x_i. During a forward pass through the network function f, the hidden state is updated and an output feature o_i is output. The encoding of a sequence using a RNN can be resumed by the following equation :

$$o_i, h_{i+1} = f(h_i, x_i).$$

As the forward pass is done iteratively in recurrent networks, the back-propagation also has to be done iteratively. This process is often referred to as back-propagation through time. A major problem of back-propagation through time is vanishing or exploding gradients [99]. It tends to prevent the network to use information about elements that are far away in a sequence (vanishing) and/or make the network training unstable (exploding). The vanishing gradient issue has been progressively improved upon [100–102] and the resulting improvements allow to better exploit long term dependencies in the sequence. Another issue with RNNs is that their training is not efficiently parallelized on modern GPUs due to the iterative forward pass and backpropagation [103].

3.3.2 Attention based architecture

As an alternative to recurrent neural networks, a more recent attention-based network architecture was introduced in [103]. Attention mechanisms were introduced in recurrent networks [104, 105] to better retain dependencies between elements independently of their distance in a sequence. The attention mechanism relies on pairwise attention coefficients $\alpha_{i,j}$ that are computed as a function of a pair of elements. These weights are often normalized [104] using a Softmax function such as $\sum_j \alpha_{i,j} = 1$.

In [103], the authors propose the transformer, an architecture entirely based on attention layers. In this architecture, a stack of attention layers encode an input sequence $\{x_i\}_{i=0}^{N}$. An one layer attention based encoder can be characterized by the following equation :

$$o_i = \sum_j \alpha_{i,j} x_j.$$

where $\alpha_{i,j} = f(x_i, x_j)$ with f a neural function. Note that unlike in RNNs, there is no iterative process so these attention based layers can efficiently be parallelized on GPUs, leading to huge time gain when training compared to RNNs.

A characteristic of attention based layers is that they are order invariant in the sense that the ordering of the input sequence does not affect the output. In tasks such as unordered pointcloud processing [106], features should be invariant to the point ordering so the lack of order is a bonus. When considering temporal data such as human motion however, order does matter. To allow the attention layers to reason on sequence ordering, a positional encoding signal is added (or concatenated) to the input sequence. For instance, in the original implementation of the transformer [103], the authors add a multivariate sinusoidal signal to the input elements where the multivariate sine function is a function of the positional index of the element in the sequence.

4

A structured latent space for human motion generation

Résumé

Dans ce chapitre, nous étudions l'apprentissage d'un espace latent structuré pour représenter et générer des mouvements 4D du corps humain denses dans le temps et dans l'espace, où un seul point d'un espace latent de faible dimension représente une séquence de plusieurs images de maillages 3D denses.

Cet espace latent est structuré de manière à ce que les mouvements similaires forment des groupes. Il intègre également des variations de durée dans le vecteur latent, ce qui permet à des séquences sémantiquement proches qui ne diffèrent que par leur déroulement temporel de partager des vecteurs latents similaires. Nous démontrons expérimentalement les propriétés structurelles de notre espace latent et montrons qu'il peut être utilisé pour générer des interpolations plausibles entre différentes actions. Nous appliquons également notre modèle à la complétion de mouvements humains en 4D, montrant ses capacités prometteuses à apprendre les caractéristiques spatio-temporelles des mouvements humains.

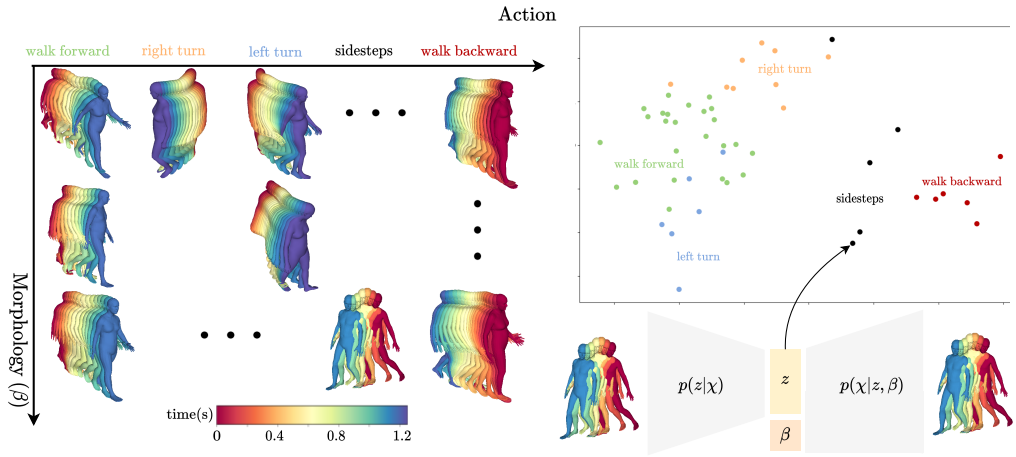


Figure 4.1: We learn a latent motion space from multi-frame 4D sequences. Left: Training sequences consist of different motions performed by different subjects (color-coded as shown in legend). Bottom right: Encoder-decoder architecture learns a latent motion space that encodes motion sequence χ into latent vector z; the decoder conditions z on morphology β. Top right: Structured latent space. Plot shows subset of 51 motions, manually labelled by action, in 2D projection of latent space. Actions form clusters.

4.1 Introduction

In this Chapter, we investigate learning a structured latent space to represent and generate temporally and spatially dense 4D human body motion, where a single point of a low-dimensional latent space represents a multi-frame sequence of dense 3D meshes. Recently, several works have proposed to learn such motion priors for 4D human body sequences of arbitrary motion by capturing information about pose changes over time [61, 66, 67], in the case of fixed sequence duration. Here, we investigate an orthogonal scenario which models sequences of *varying duration*, by considering motions sufficiently similar to allow temporal alignment.

Learning a generative model of 3D human motion of varying duration with a structured latent space is of interest for a wide set of applications in computer vision and graphics, where a lightweight 4D representation translates to gains in information processing. By capturing a spatio-temporal motion prior, the model opens new directions for many completion tasks given temporally, geometrically sparse or incomplete inputs, as it allows to reason within a restricted plausible spatio-temporal solution space.

Learning this space is a difficult task with two major challenges. First, the model needs to capture the intertwined variations of different factors, *e.g.* morphology, global motion, body pose, and temporal evolution of the motion, and do so for motions that differ in duration. In particular, while it is known that morphology impacts the way a motion is performed [12, 107], it remains challenging to take this correlation into account during motion generation. Second, the amount of data that needs to be processed for training is large, as typical acquisition systems for dense human body motions produce $30 - 50$ frames per second, with each frame containing thousands of geometric primitives.

To address these challenges, we take inspiration from two existing lines of work. The first studies temporally dense skeletal data, with the goal of generating skeletal human motion sequences that capture the temporal evolution of the global motion [11, 108, 109]. These do not address dense surfaces. The second line of work represents realistic 3D human body surfaces in a low-dimensional shape space [9, 48], but do not consider the temporal dimension.

We combine the advantages of both in a data-driven framework that learns a latent motion representation, which allows to simultaneously represent temporal motion information and detailed 3D geometry at every time instant of the motion. The learning uses multi-frame sequences as input and output. Inspired by works on morphable body models [12, 43], we align the training sequences both temporally and spatially, which leads to comparisons at corresponding instances of the motion and anatomically corresponding points. In particular, we consider motions whose duration vary significantly while geometrically similar enough to allow for temporal alignment. These motions are performed by actors in minimal clothing to allow for effective spatial alignment.

In our experiments, we consider motions during which the hip performs a cycle, as this includes common motions such as walking and running, and generalizes to more complex motions such as dancing or jumping jacks, while imposing no constraints on the arm movements. The resulting latent space is verifiably structured, and allows generating plausible interpolations between different types of locomotion that outperform linear and per-frame interpolation baselines. As illustrated in Fig. 4.1, our motion space also learns the interaction between morphology and motion, as generating motions with the same point in latent space conditioned on different representations of morphology leads to motion differences that confirm findings in prior studies conducted on sparse motion data [12].

Our model can serve as prior to complete both spatially and temporally sparse sequences. Given as input unmatched and temporally incoherent point clouds sparsely sampled in space or time, accurate complete 4D reconstructions are obtained. For spatio-temporal completion, our method outperforms state-of-the-art motion priors that encode human motion sequences of fixed duration [66, 67] with sufficient temporal samples, in spite of being trained on significantly fewer data. It also outperforms a state of the art spatial completion baseline when few samples are available [110].

In summary, we make the following major contributions. First, we present a latent motion space that allows representing and generating multi-frame sequences of dense 3D meshes of varying duration, which accounts for interaction between morphology and motion. Second, we demonstrate that this latent space is structured: similar motions form clusters, and linear interpolation in latent space outperforms baselines. Third, when using our motion space as prior, we outperform state of the art for the application of motion completion from sparsely sampled data in space or time. The code of this chapter is available at https://gitlab.inria.fr/mmarsot/a-structured-latent-space

4.2 Generative model of multi-frame sequences

Two previously identified major challenges need to be tackled in our model: first the very large dimensionality of the problem as is concerns temporally dense sequences of dense 3D meshes; second the modeling of intertwined variations in the generation of 4D sequences, between subject morphology, motion, and temporal unfolding.

To address them, we first need to ensure that we produce a compact and structured motion representation. Our general strategy for this is to extend the static shape space representations (*e.g.* SCAPE, SMPL) to the spatio-temporal domain, with a similar low-dimensionality characteristic, as detailed in Section 4.2.1. Second, we articulate our data-driven strategy around an encoder-decoder architecture (Section 4.2.2). Notably, to explicitly model the interaction between morphology and motion, we adopt a disentangled latent representation and choose to condition the motion generation on a representation of morphology. Third, we build our experimental demonstration in a use case that benefits from these choices, focusing our effort on a database of 4D human motion sequences that perform a cyclic motion of the hip joint. This allows to show the intended behavior for this space, which is to group similar locomotion (*e.g.* all walking motions) in clusters. Section 4.2.3 explains how the model is trained.

4.2.1 Representation of motion sequences

Fig. 4.2 (top left) shows our representation for 4D sequences. A 4D human motion sequence is parameterized by a single point z in motion space and an identity parameter β representing the morphology of the moving person.

Anchor frames To represent motion data, we align an unstructured spatio-temporal motion signal. Temporally, we uniformly sample n frames from the motion signal, which we call anchor frames in the following. These anchor frames allow representing motions of various duration with the same number of frames. Spatially, we build on 3D morphable body models to align the frames *e.g.* [10, 44, 45]. These models represent static 3D human body surfaces using a common mesh template. This results in n aligned anchor meshes, making motion comparison practical.

Representing temporal evolution The resulting anchor mesh sequence $M = [m_1, \ldots, m_n]$ does not represent the temporal evolution of a motion. The temporal sampling causes an information loss, as it is invariant to similar motions with different temporal unfolding like walking and running. Therefore, we associate to anchor mesh m_i a timestamp τ_i, and call the timestamp vector $\mathcal{T} = [\tau_1, \ldots, \tau_n]$. The representation $[M, \mathcal{T}]$ is high-dimensional. To simplify processing and disentangle the influence of morphology on motion, we leverage 3D morphable body models that decouple the influence of morphology and pose. By holding morphology constant over M, we can represent each m_i using parameter vectors for morphology β, pose θ_i, and global translation γ_i. While any decoupled static model can be used, *e.g.* [41, 50, 51], in our implementation we chose the commonly used SMPL model [10] as the AMASS dataset [6] is parameterized by SMPL. We denote the model function by $SMPL$ such that $m_i = SMPL(\theta_i, \gamma_i, \beta)$ and thus $M = [(SMPL(\theta_0, \gamma_0, \beta), \ldots, SMPL(\theta_n, \gamma_n, \beta)]$. By denoting the pose and global translation vectors by $\Theta = [\theta_1, \ldots, \theta_n]$ and $\Gamma = [\gamma_1, \ldots, \gamma_n]$,

respectively, $[\Theta, \Gamma, \beta, \mathcal{T}]$ is a low dimensional representation of $[M, \mathcal{T}]$. To retain variation in global displacement (*e.g.*walking backward or forward) and temporal evolution (*e.g.*walking or running), we model Γ and $\mathcal{T}$ in the multi-frame sequence representation. τ_i allow placing freely and on any time span length the anchor meshes, thereby allowing to represent motions with various duration using a constant number of meshes.

Notation To emphasize the difference between motion and morphology parameters, we denote $\chi = [\Theta, \Gamma, \mathcal{T}]$ the motion parameters and introduce function $\mathcal{F}$ such that $[M, \mathcal{T}] = \mathcal{F}([\chi, \beta])$. As pre-processing for training, we map a raw motion sequence to the SMPL mesh template using existing solutions [6, 111]. Let $SMPL^{-1}$ denote the mapping function which associates a single raw motion frame to its representation parameters θ, γ, β.

Numerical representation In practice, we represent β and Γ as in SMPL. Pose features Θ are joint rotations of a skeleton, represented by a continuous 6D rotation [112] based on a Graham-Schmidt orthogonalization of 3D rotation matrices. That was shown to outperform other rotation representations when training neural networks.

4.2.2 Architecture

We assume our data distribution $p(\chi)$ to be explained by two independent latent variables z, β such as : $p(\chi) = \int \int p(\chi|z, \beta) dz d\beta$.

To learn the interaction between morphology and motion patterns, we condition motion generation on β using an architecture based on conditional variational auto-encoders (CVAE) [113], as shown in the bottom of Fig. 4.2. Our architecture encodes motion vector χ into a low-dimensional latent vector z, and β is used as condition for the decoder, thereby allowing to capture dependencies between χ and β. We assume z and β to be independent and learn a disentangled representation. Therefore, the encoder models posterior distribution $p(z|\chi)$, and is not conditioned on β.

The encoder outputs are interpreted as mean μ and standard deviation σ of the posterior distribution of the latent space. The corresponding latent vector z is sampled as $z = \mu + \epsilon \times \sigma$, with $\epsilon \sim \mathcal{N}(0, 1)$. We denote the probabilistic encoding function by $E : \chi, \epsilon \mapsto z$, and the decoding function as $D : z, \beta \mapsto \hat{\chi}$. The decoder takes (z, β) as input, and outputs $\hat{\chi} = [\hat{\Theta}, \hat{\Gamma}, \hat{\mathcal{T}}]$ which are converted back to a sequence of timestamped anchor meshes $[\hat{M}, \hat{\mathcal{T}}] = \mathcal{F}(\hat{\chi}, \beta)$. To go from a reconstructed sequence $\hat{M}$ to a temporally continuous motion, we assume constant motion between anchor meshes.

4.2.3 Training

The network is trained with a reconstruction term to minimize the difference between the input and output vectors, and a regularization term to constrain the latent variables to follow a known prior distribution. The training is divided into two phases. First, we consider a reconstruction loss on χ to allow for fast and memory efficient initialization. Second, we replace it by a loss computed directly on the sequence of anchor meshes M in $\mathbb{R}^3$.

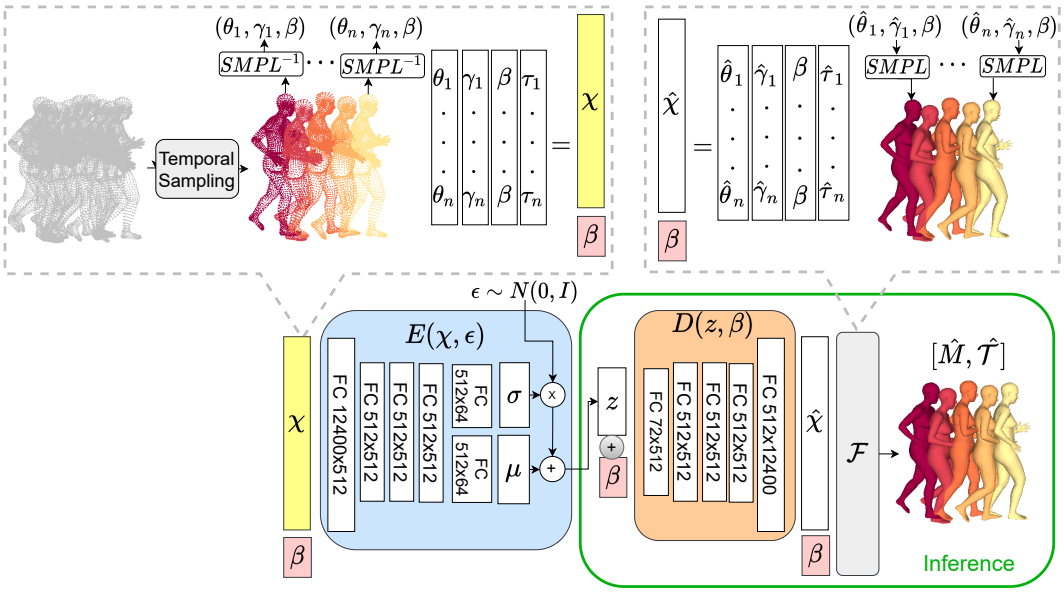


Figure 4.2: Overview of motion representation and architecture. Top: representation. Left: pre-processing during training samples n anchor frames and extracts per-frame representations of pose θ, translation γ and morphology β with their timestamp τ to obtain motion representation χ and morphology β. Right: illustration of the function $\mathcal{F}$. Bottom: our architecture consists of a probabilistic encoder E and a decoder D, and learns a mapping from χ to a single latent vector z. At inference time, D conditions z on β to generate sequence features $\hat{\chi}$ (green box).

Reconstruction loss on χ The standard reconstruction term would be $(\hat{\chi} - \chi)^2$. To balance the influence of the different types of information captured by χ, we divide this loss into three terms operating on pose $\mathcal{L}_{pose} = (\Theta - \hat{\Theta})^2$ translation $\mathcal{L}_{trans} = (\Gamma - \hat{\Gamma})^2$, and time $\mathcal{L}_{time} = (\mathcal{T} - \hat{\mathcal{T}})^2$. This gives a total reconstruction loss

$$\mathcal{L}_{rec} = \omega_{pose}\mathcal{L}_{pose} + \omega_{trans}\mathcal{L}_{trans} + \omega_{time}\mathcal{L}_{time}, \tag{4.1}$$

where $\omega_{pose}, \omega_{trans}$ and ω_{time} are the respective weights of the partial reconstruction losses. To minimize $\mathcal{L}_{rec}$, we use adaptive weights to trade off the relative influence of $\mathcal{L}_{pose}, \mathcal{L}_{trans}$ and $\mathcal{L}_{time}$ [114], which do not have the same order of magnitude. Adaptive weights are initialized at 1.0 and updated automatically during training, which ensures that the partial losses are decreasing in similar proportions.

Reconstruction loss in 4D The second reconstruction loss is $\mathcal{L}_{spatial} = (M - \hat{M})^2$, where M denotes the 3D coordinate vector or the anchor mesh sequence, resulting in the $4D$ reconstruction term

$$\mathcal{L}_{rec4D} = \omega_{spatial}\mathcal{L}_{spatial} + \omega_{time}\mathcal{L}_{time}, \tag{4.2}$$

where $\omega_{spatial}$ is an adaptive weight.

Regularization loss The regularization term is the squared Kullback-Leibler (KL) divergence between the learned posterior distribution $\mathcal{N}(\mu, \sigma)$ of the latent variable z and a normal prior distribution $\mathcal{N}(0, 1)$, denoted $\mathcal{L}_{KL}$.

Optimization A common problem when training VAEs is the weighting of the regularization loss versus the reconstruction loss. We use a fixed weight $\omega_{KL} =$

0.01 to trade off these losses. The training optimizes first

$$\mathcal{L}_{init} = \mathcal{L}_{rec} + \omega_{KL}\mathcal{L}_{KL} \tag{4.3}$$

and subsequently

$$\mathcal{L} = \mathcal{L}_{rec4D} + \omega_{KL}\mathcal{L}_{KL}. \tag{4.4}$$

4.2.4 Implementation Details

In our motion representation χ, we do not consider the SMPL components related to hands or dynamic components available in AMASS. We further discard the two foot joints because they have constant rotation. This leaves a total of 20 joints. Our representation χ consists of 100 timestamped anchor meshes, each of which is represented by 124 parameters (120 for θ, 3 for γ and 1 for τ). 100 anchor meshes are chosen as they provide a good trade-off between the error introduced by the sampling and the dimensionality of χ. To normalize the data, we normalize the translation γ in $[-1, 1]^3$, and the timestamps τ in $[0, 1]$ using min max scaling over the training set. We remove the identity rotation $[1, 0, 0, 0, 1, 0]$ from the 6D representation, which leads to a significant gain in reconstruction accuracy compared to the classic scaling $\frac{\theta - \mu_\theta}{\sigma_\theta}$ due to some components exhibiting a standard deviation close to zero.

We train for 5000 epochs with $\mathcal{L}_{init}$, using a learning rate of $1e^{-3}$ and a batch size of 256. Each epoch takes 6 s for a total training time of 8 hours. We train with $\mathcal{L}$ for 200 epochs using a smaller batch size of 16 for memory reasons and a learning rate of $1e^{-4}$. Here epoch time is 8 min for a total training time of one day. The training is done on an NVIDIA Quadro RTX8000 with 48G of GPU RAM. We use $\omega_{kl} = 0.01$ for both steps and chose a latent dimension for the motion space z of 64, and of 8 for β. Note that mesh vertex positions are in meters during training. We initialize all dynamic weights to 1.0 and GradNorm [114] updates the weights dynamically.

4.3 Evaluation

This section investigates the influence of the latent space dimension and regularization and presents comparisons to baselines. We also investigate the structure of the learned latent space by visualizing labeled motion sequences in latent space and by linearly interpolating between pairs of input motion sequence. Finally, we demonstrate that the proposed model learns information on the interaction of morphology and motion by visualizing the motion changes caused by changing β for a fixed point z.

4.3.1 Data

We automatically extract motion sequences during which the hip performs a cycle from a dataset by comparing all subsequences to a set of 4D template motions using dynamic time warping [115] as distance. Subsequences are considered if

this distance is below a threshold. As post-processing, we prune segments with a duration above $3s$ or below $0.3s$. We manually generate two 4D template motions as gait cycles starting with the left and right foot.

We experiment with AMASS [6] and Kinovis [111] datasets. AMASS regroups a large set of MoCap recordings and fits a parametric body model to all data. When splitting the AMASS dataset into training and test sets, we treat all sequences emanating from the same MoCap dataset as one entity. We leave the MoCap datasets "MPI_mosh", "SFU", and "TotalCapture" for testing, and call this dataset *AMASS test set*. For training, our cropping results in 12085 sequences corresponding to $\approx 4.5h$ of motion. The Kinovis dataset contains 4D motion sequences captured using a multi-view platform and allows to evaluate the generalization of the model to densely captured 4D data. We consider all walking and running sequences, pre-process the data by fitting SMPL before extracting cyclic hip motions, and call this dataset *Kinovis test set*. This results in 37 test sequences, some of which contain less than 100 frames; we augment shorter sequences to 100 frames using linear interpolation between the 6D rotations.

To allow for efficient learning, the sequences are spatially aligned by zeroing the initial translation, and we use the identity rotation as initial rotation of the root joint to be invariant in the ground plane.

4.3.2 Influence of latent space dimension and regularization

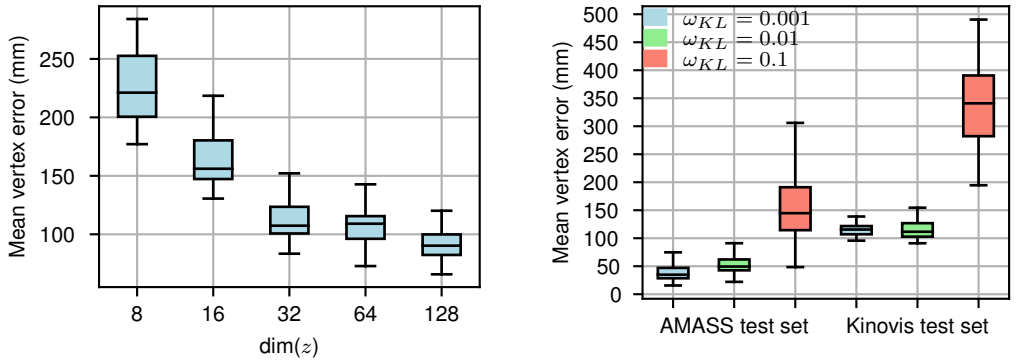
We now investigate the influence of the latent space dimension and regularization on the quality of the model. To evaluate the model's quality, we measure its reconstruction error, which characterizes the model's ability to reconstruct examples unseen during training and is defined as

$$\frac{1}{nk}\left(M - \hat{M}\right)^2, \tag{4.5}$$

with $\left[\hat{M}, \hat{\mathcal{T}}\right] = \mathcal{F}(\hat{\chi}, \beta) = \mathcal{F}(D(E(\chi, \epsilon), \beta), \beta)$, where n is the number of anchor frames and k the number of vertices per frame. As second qualitative error measure, we consider the model's ability to allow for the generation of plausible new sequences by sampling in latent space. In practice, we consider samples that are linearly interpolated between sequences of the test set.

Latent space dimension We first study the influence of the dimensionality of the latent space on the model quality. Fig. 4.3a shows the impact of the dimension of z on the reconstruction error on the AMASS test set. As expected, the bigger the latent space dimension, the smaller the error. However, for $dim(z) > 64$, the error starts to stagnate. Therefore, we set the dimension of the latent space to 64.

Latent space regularization The regularization of the latent space has a major impact on the model quality. It is controlled by coefficient ω_{KL}, which weighs the influence of latent space regularization at the cost of reconstruction accuracy. Fig. 4.3b shows the reconstruction error on models trained with different values for ω_{KL}. The smaller ω_{KL}, the smaller the reconstruction error. However, with $\omega_{KL} = 0.001$, the model no longer allows generating plausible interpolations. Therefore, we set $\omega_{KL} = 0.01$ in the following.



(a) Influence of dimension *(b) Influence of regularization*

Figure 4.3: Influence of latent space dimension and regularization on reconstruction error. Left: Increasing dimension of the latent space leads lower reconstruction errors on AMASS test set. Right: Smaller regularization ω_{KL} leads lower reconstruction errors on both test sets. Boxes follow [2].

4.3.3 Comparison to baseline models

We compare our model to two baselines *w.r.t* the reconstruction error $\frac{1}{nk}(M - \hat{M})^2$, with $\left[\hat{M}, \hat{\mathcal{T}}\right] = \mathcal{F}(\hat{\chi}, \beta) = \mathcal{F}(D(E(\chi, \epsilon), \beta), \beta)$, where n is the number of anchor frames and k the number of vertices per frame. The first baseline applies a linear principal component analysis (PCA) to our representation $[\chi, \beta]$, thereby evaluating the value of using a non-linear model. PCA has access to morphology information when projecting the motion representation to latent space, and reconstructs both $\hat{\chi}$ and $\hat{\beta}$. To provide a fair comparison, we consider the original β instead of $\hat{\beta}$ in PCA reconstructions and set the PCA latent dimension to $dim(z) + dim(\beta)$ with $dim(z) = 64$ and $dim(\beta) = 8$. The second baseline considers our model after optimizing $\mathcal{L}_{init}$ only, which operates on skeleton representations, thereby evaluating the value of learning from data that is densely sampled in space.

Fig. 4.4 shows reconstruction errors for the different models. While PCA provides low reconstruction errors, these are further improved using our model. Our model also improves over its initialization, which shows that considering densely sampled data significantly impacts performance.

4.3.4 Motion space structure and interpolation

Fig. 4.1 illustrates that our model learns a latent space in which sequences of similar actions are clustered. For the purpose of visualization, we labeled 51 motions by actions and assigned a unique color per action. These motions are then encoded into latent space, which is linearly reduced to two dimensions. Points of the same action form clusters.

This structured latent space can be exploited to generate plausible interpolations between input motions using linear interpolation. Given start and target motion sequences as input, we encode them as (z_s, β_s) and (z_t, β_t), and generate interpolating motion sequences by decoding $((1-k)z_s + kz_t, (1-k)\beta_s + k\beta_t)$ at intermediate position $k \in [0, 1]$.

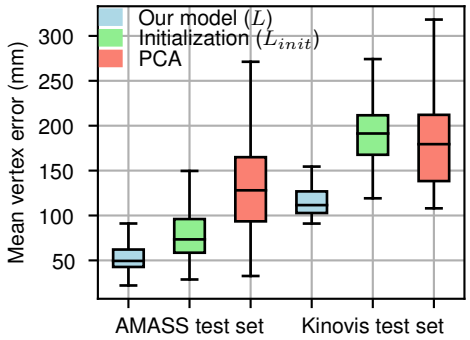


Figure 4.4: Comparison to baselines w.r.t. reconstruction error. Our model (blue) outperforms a linear PCA baseline (red) and a baseline that considers spatial sampling at skeleton level (green). Boxes follow [2].

We compare our results to two baselines. The first uses the PCA model from the previous section and linearly interpolates in PCA space. This comparison, called PCA, evaluates the value of using a non-linear model. The second baseline operates per anchor frame and interpolates linearly between the global displacements, time stamps and morphology parameters, and with spherical linear interpolation [116] (SLERP) between skeletal poses. This comparison, called SLERP, evaluates the value of learning a motion model instead of operating independently per-frame. For all interpolations, visualizations show $k = 0.5$. In the following, we interpolate between sequences that differ in each of the factors encoded in χ.

Interpolating sequences of different duration To inspect temporal information learned by our model, we interpolate between a running and a walking motion. For our model, the duration of the intermediate sequences monotonically decreases when going from running to walking, and the intermediate sequences are realistic as shown in Fig. 4.5(a), showing that our motion space has captured information on the temporal evolution τ. PCA and SLERP baselines also lead to plausible interpolations.

Interpolating sequences of different global displacement To inspect global displacement, we interpolate between a forward and a backward walk. Our intermediate sequence corresponds to a tiny step, shown in Fig. 4.5(b). There were no steps this small in the training set. PCA and SLERP baselines fail to interpolate global translation realistically, resulting in foot skating.

Interpolating sequences of different pose To inspect the learned information of pose, we consider global and articulated pose separately. First, we interpolate between sequences of turning left and turning right while walking, exhibiting mostly global pose change. The intermediate sequences using our model gradually change from a left to a right turn as shown in Fig. 4.5(c). PCA and SLERP baselines fail due to the ambiguity when interpolating between opposite rotations, while our model leverages spatio-temporal information to alleviate this ambiguity. Second, we interpolate between walking and walking while carrying an object on the head, exhibiting mostly articulated pose change. The intermediate sequence with our model results in realistic intermediate positions for the arms, gradually elevating them to head level as shown in Fig. 4.5(d). Both baselines lead to plausible

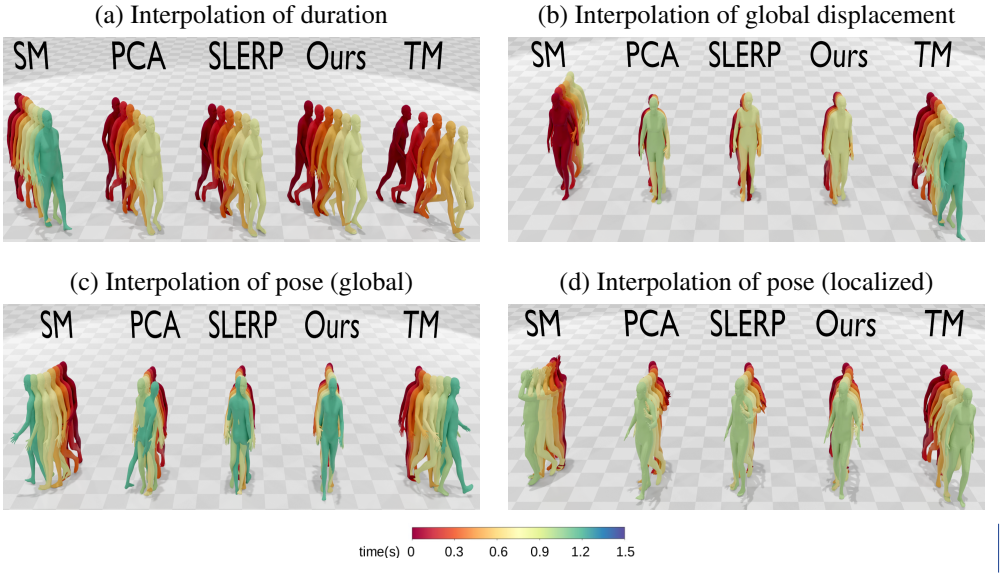


Figure 4.5: Linear interpolations in latent motion space. Each figure left to right : starting motion, PCA interpolation, SLERP interpolation, our interpolation, and target motion. Sequence models are rendered with a color-coded frame time. ***(a)*** *Running & walking.* ***(b)*** *Walking backward & forward.* ***(c)*** *Left & right turn.* ***(d)*** *Walk & walk carrying an object on the head. All interpolations with our model are plausible, while baselines fail in (b) and (c).*

interpolations.

In summary, while our model generates visually plausible interpolations for all parameters encoded in χ, both baselines exhibit failure cases in some scenarios, which shows the value of learning a non-linear 4D motion model.

4.3.5 Interaction between morphology and motion

To examine the influence of morphology β on 4D motion χ, we consider a fixed jogging motion represented by z^* in motion space and visualize χ when setting β to ± 3 standard deviations along the first and second principal components. To understand the subtle motion differences, we further visualize the spatio-temporal gradient $\frac{\partial D(z^*,\beta)}{\partial \beta}$ at $\beta = 0$, *i.e.* we look at the gradient learned by the decoder *w.r.t* morphology at the mean shape.

We compare our result to a baseline that uses the initial pose parameters and β to reconstruct a dense 3D body model using SMPL per frame. This evaluates the influence of learning the interaction between morphology and motion.

Fig. 4.6 shows the impact of the first (left) and second (right) principal components of β. The top row shows a color coding of the gradient learned by our decoder *w.r.t* β on the 4D sequence, and the middle row shows the corresponding 4D motions obtained by our model. The bottom row shows the result of the baseline color-coded by the distance to the result of our model. Changing the first principal component impacts perceived gender. For our model, this changes the 4D motion on the right shoulder and left hip, in agreement with prior studies showing that shoulder sway and hip motion are statistically gender related [12]. Changing the second principal component leads to perceived weight change. For our model, this impacts

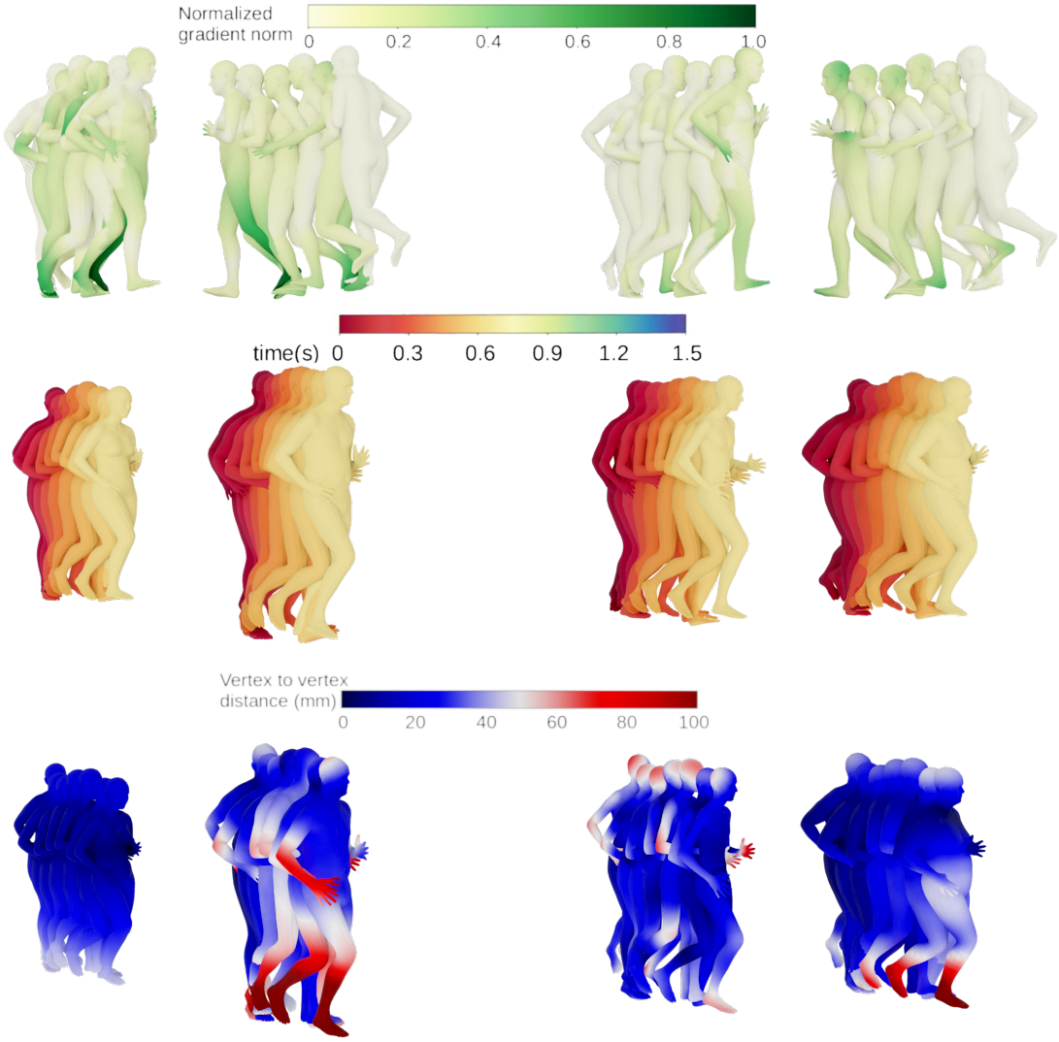


Figure 4.6: Interaction between morphology and motion on 1^{st} (left) and 2^{nd} (right) principal components of β. Top: visualization of our decoder's normalized gradient w.r.t β. Middle: our inferences with fixed latent motion vector and β taken at ± 3 std. deviations. Bottom: baseline per-frame motion transfer using SMPL for same fixed motion and β taken at ± 3 std. deviations, color coded by per-vertex distance to our result. Our learned correlation has significant impact on motion, which differs up to $10cm$ from baseline.

the 4D motion at the right arm, head and neck. The spatio-temporal areas affected by our motion model are the ones where the baseline leads to significantly different results with up to $10cm$ distance. This shows that our model learns meaningful interactions between morphology and motion.

4.4 Application to motion completion from spatio-temporally sparse input

This section applies our model to spatio-temporal completion, which has applications ranging from the registration of a raw spatio-temporally densely scanned 4D sequence over computing realistic in-betweenings for a set of frames sparsely sampled in time to completing full human body motion from a sparse set of MoCap markers.

4.4.1 Completion methodology

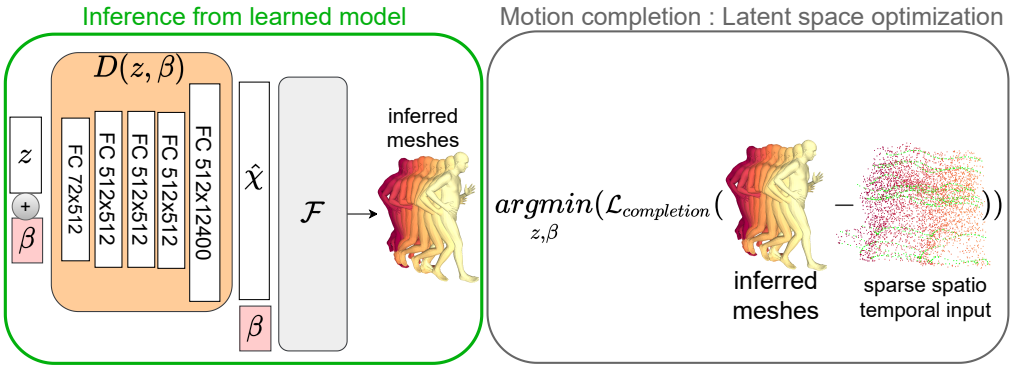


Figure 4.7: Motion completion. We minimize a loss w.r.t latent representation (z, β). Left: inference pipeline. Right: we optimize $\mathcal{L}_{completion}$ between a sparse 4D point cloud and inferred meshes.

We consider as input partial motion sequences of unordered dense 3D scans with possibly additional synchronized MoCap for k landmarks and associated time stamps. Let $S = [s_1, \ldots, s_n]$ denote a sequence of n anchor scans uniformly sampled in time, $L = [l_1, \ldots, l_n]$ the corresponding synchronized sequence of landmarks, and $\mathcal{T} = [\tau_1, \ldots, \tau_n]$ the corresponding time stamps. Some anchor frames are empty, and our input consists of a set I of frame indices i for which s_i or l_i and τ_i are given.

To compute a sequence of anchor meshes $\hat{M}$ with associated time stamps $\hat{\mathcal{T}}$ that approximate the input, we decode a full sequence of anchor frames $[\hat{M}, \hat{\mathcal{T}}]$ using $\mathcal{F}\left(D(z,\beta),\beta\right)$ and optimize for latent vectors z^*, β^* as

$$z^*, \beta^* = \underset{z,\beta}{argmin}(\mathcal{L}_{completion}(\hat{M}(z,\beta), \hat{\mathcal{T}}(z,\beta), S, L, \mathcal{T})), \tag{4.6}$$

where

$$\begin{aligned}\mathcal{L}_{completion} &= \omega_{dense} \sum_{i \in I} \text{Chamfer}(\hat{m}_i(z,\beta), s_i) \\ &+ \omega_{mocap} \sum_{i \in I} \text{Landmark}(\hat{m}_i(z,\beta), l_i) \\ &+ \omega_{time} \sum_{i \in I} (\hat{\tau}_i(z,\beta) - \tau_i)^2. \end{aligned} \tag{4.7}$$

The weights ω_{dense}, ω_{mocap} and ω_{time} are adaptive [114]. When $s_i = \emptyset, \omega_{dense} = 0$ and when $l_i = \emptyset, \omega_{mocap} = 0$. Varying ω_{mocap} allows to evaluate the benefit of having tracked input markers. Chamfer is the Chamfer distance between two point clouds and Landmark is the squared Euclidean distance between k vertices of the SMPL template, selected once for all experiments, and the k given landmarks. This optimization is visualized in Fig. 4.7.

4.4.2 Completion dataset : CHUM

We introduce a new dataset of cyclic human motion (CHUM), which was captured using a 4D modeling platform with 68 RGB cameras and a Qualisys MoCap system. Data consists of dense scans of approximately 10000 points acquired at 50fps with synchronized MoCap for 16 markers. We recorded 4 actors in tight clothing with different morphologies (2 males and 2 females). The actors were volunteers that provided their informed consent. Each actor performed the following motions :

- Walking : forward, backward, clockwise, counter-clockwise
- Running : forward, clockwise, counter-clockwise
- Sidestepping : left to right and right to left
- Skipping forward
- Football kick
- Boxing while moving forward
- Cartwheel (1 actor only)

These motions were chosen to evaluate our method with respect to different factor of variations.

Global displacement variations These motions exhibit variability in the global displacement with three categories of linear trajectories : forward, backward, sidesteps and two categories of non-linear trajectories : clockwise and counter-clockwise.

Pose variations The motions exhibit a range of challenging poses for both upper and lower body parts with the boxing motion being especially challenging in terms of upper body pose and the skipping/ kicking motion being especially challenging in terms of lower body pose. The cartwheel is also challenging in terms of global body orientation.

Temporal unfolding variations The running sequences are faster and provide variability in the temporal unfolding of the motion.

In terms of morphology, the actor have significantly distinct body shapes. With significantly different body heights and body mass index.

In this Chapter, we consider 4 segments of each original sequence apart from the cartwheel which does not exhibit cyclic hip motion to evaluate our method. As pre-processing, we estimated an initial 3D transformation (rotation + translation) to align each segment at $t = 0$. We do not need to fit SMPL to the dense scans because our loss function $\mathcal{L}_{completion}$ does not require correspondence information.

4.4.3 Results

We compare our results to three state-of-the-art approaches. The first performs static 3D completion per frame [110]. Due to its high computational complexity, we apply the static method to a subset of CHUM while other methods are applied to the full dataset. This method is only applicable for spatial completion where observations are available at every frame. The second and third are motion spaces for sequences of fixed duration that can serve as prior [66, 67]. Given a partial motion as input, we optimize a latent motion vector z, a morphology β and a set of per-frame translation parameters for [67], as global translation is not encoded in this motion space. In case of temporally sparse input, translation parameters are only optimized for frames in I and the remaining are found using linear interpolation between the closest observed frames. For [67] and [66], we optimize for $\mathcal{L}_{completion}$ with $\omega_{time} = 0$, as these motion spaces are designed for sequences of fixed duration and cannot benefit from time stamp information. These methods are applicable for both spatial and temporal completion. [67] uses a latent space of 256 dimensions while [66] uses a total of 36. For fair comparison to the more precise method, we re-train our model with $dim(z) = 256$.

Table 4.1: Comparative evaluation of motion completion. Mean and standard deviation of Chamfer distance in mm, computed between completions and ground truth anchor scans from CHUM. N.A. means not applicable.

	Points per scan p					Frames (f)		
	0	50	100	1000	10000	5	20	100
Ours (dim(z)=256)	**42±48**	**23±7**	**21 ±9**	**20±10**	20±10	30±14	**20±10**	**20±10**
[110]	N.A.	58±0.99	47 ±0.6	21±0.3	**10±0.46**	N.A.	N.A.	N.A.
[67]	46 ± 52	26±8	24 ±9	22±10	22±10	**22±10**	22±11	22±10
[66]	216 ± 41	88±10	69 ±10	36±10	26±11	33±13	26±11	26±11

Spatial completion We first evaluate the quality of spatial completion by simulating different levels of spatial sparsity by varying the number of points p per scan s_i. The sampled points are not in correspondence over time. Table 4.1 shows the evolution of the reconstruction error in mm when varying p. Our method outperforms the static method [110] for very sparse scans ($p < 100$), the two methods are on-par for denser scans ($p = 1000$), and the static method outperforms our method for dense scans ($p = 10000$). This quality on sparse scans is achieved because our model optimizes for all frames simultaneously, so few points per scan suffice to find a plausible solution. The static method deforms a template, and

can capture higher levels of geometric detail for dense scans. Our method further outperforms state-of-the-art motion spaces [66, 67], in spite of being trained on significantly less motion data (4.5h for ours vs. 34h for [66, 67]). A qualitative comparison for the spatial completion task with $p = 100$ and available landmark data is shown in Fig. 4.8. Note that our method leads to more plausible wrist and hand motion than [67] , better temporal coherence than [110], better leg motion than [110] and [66] and better global translation than [66].

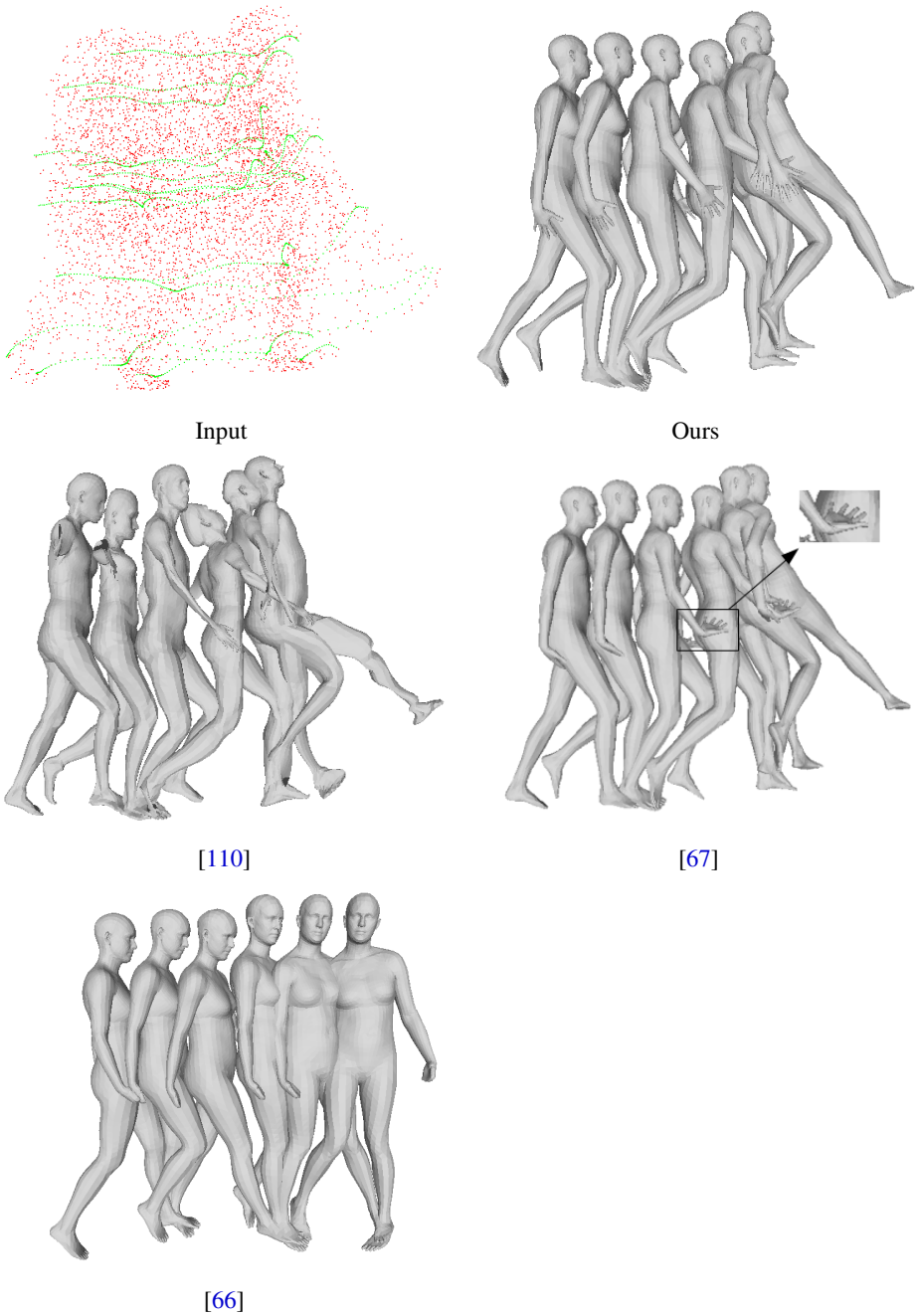


Figure 4.8: Qualitative comparison of spatial completion on kick sequence from CHUM with $p = 100$. Input scans shown in red, landmarks in green. Visualization shows 6 of 100 completed frames. Note that our motion completion is plausible and coherent with input.

Temporal completion Second, we evaluate the quality of temporal completion by varying the number of observed frames. To vary this number for each test sequence, we reduce I to simulate lower frame rates. Table 4.1 shows the evolution of the reconstruction error. The $f = 100$ frame completion task includes all frames and is given as reference. The model extrapolates with almost no loss of precision with $I_{20} = [5, 10, \ldots, 95, 100]$ (20 frames) and the error is still low with $I_5 = [20, 40, 60, 80, 100]$ (5 frames). While the motion space for sequences of fixed duration [67] is better for sparsely sampled temporal data, we outperform both [67] and [66] for temporally denser data, in spite of using significantly less training data.

4.5 Conclusions

This Chapter presented a latent space that allows to represent and generate multi-frame sequences of human motion in 4D. This latent space contains information on global motion, body pose, temporal evolution of the motion, and morphology. We demonstrated that similar motions tend to form clusters in this latent space and that linear interpolations between pairs of sequences in latent space are plausible. Furthermore, our model to generate 4D motion sequences captures the interaction between morphology and motion. We applied this model to spatio-temporal motion completion, demonstrating state-of-the-art performance.

This first representation however exhibits two limitations. Firstly, the temporal alignment restricts it to motions exhibiting a cyclic hip movement. Secondly, it only characterizes a temporal segment of motion. In Chapter 5, we propose a sequential representation to represent longer term and more general motion. We additionally improve the model to represent motion as a continuous function of time, allowing for more flexibility with the temporal resolution.

5

Sequence of latent primitives for generic long motion modelling

Résumé

Dans ce chapitre, nous proposons une représentation séquentielle qui représente de longues séquences de 5 secondes et plus, échantillonnées à des fréquences d'images arbitraires, sans restriction sur le mouvement effectué. L'idée principale est que le mouvement est mieux représenté par une séquence de primitives latentes que par un seul vecteur latent.

Nous montrons expérimentalement que cette représentation séquentielle a une meilleure capacité de généralisation qu'une référence utilisant un seul code latent, et qu'elle généralise à des durées du mouvement en dehors de l'ensemble d'apprentissage. De même que la méthode présentée au chapitre 4, cette représentation peut être exploitée pour compléter des données humaines non structurées échantillonnées de manière éparse dans l'espace et le temps en encodant les données partielles dans notre espace de représentation. Nous démontrons que cette approche fournit des résultats de pointe pour une tâche de complétion spatio-temporelle à la fois sur des données synthétiques et sur des données acquises dans un studio multi-vues.

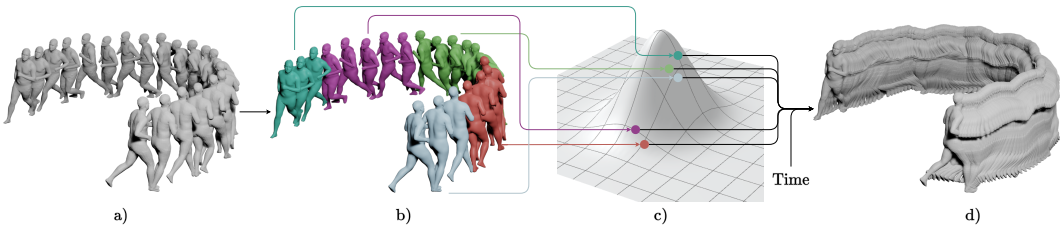


Figure 5.1: We propose a novel representation of human motion which encodes motion as a sequence of latent primitives. Given an input sequence of meshes (a), our method simultaneously learns its segmentation (b) and a latent space encoding per segment (c). The latent primitives are decoded to a temporally continuous sequence of meshes (d).

5.1 Introduction

In Chapter 4, we proposed a method that encodes motion as a single latent vector. We showed that this representation is efficient when considering temporally aligned sequences but this temporal alignment also restricted the variety of the motion. Another limitation shared by single vector latent representations [13, 67, 86, 117] is that their performance drops significantly when considering longer motion duration. In this Chapter, we propose a sequential representation that represents long sequences of 5 seconds and more, sampled at arbitrary frame rates, and no restriction on the performed motion. The key insight is that motion is better represented by a sequence of latent primitives than by a single latent vector. Making a discrete analogy between actions and latent primitives, in a dataset of x different actions, a motion prior needs a total of x^y different latent codes to represent all possible y element sequences of actions while a sequential motion prior requires only x different latent codes. As y increases with longer motions, the number of latent codes grows exponentially with motion duration for existing motion priors but remains constant in a sequential representation.

The model presented in this chapter simultaneously learns a segmentation and a latent space encoding per segment, as illustrated in Fig. 5.1. We leverage a sequence-to-sequence (seq2seq) architecture based on [103] that is flexible *w.r.t* the sequence length of the input and allows to directly encode the motion into a sequence of latent primitives. We then decode the sequence of latent primitives using a decoder implicit in time that outputs a parametric 3D human body model for any given time instant.

We show experimentally that this sequential representation has better generalization capacity than a baseline using a single latent code, and generalizes to motion duration outside the training set. Similarly to the method presented in Chapter 4, this representation can be leveraged to complete unstructured human data sampled sparsely in space and time by encoding the partial data into our representation space. We demonstrate that this approach provides state-of-the-art results for a spatio-temporal completion task on both synthetic data and data acquired in a multi-view studio.

In summary, this Chapters contributions are

- A novel motion representation using a sequence of latent primitives.
- An implicit representation of the temporal dimension allowing for flexible temporal resolution.
- An ablation and a comparative study showing significant improvement *w.r.t* existing motion representations.

The code for this chapter is available at https://gitlab.inria.fr/mmarsot/new_segmentation

5.2 Overview

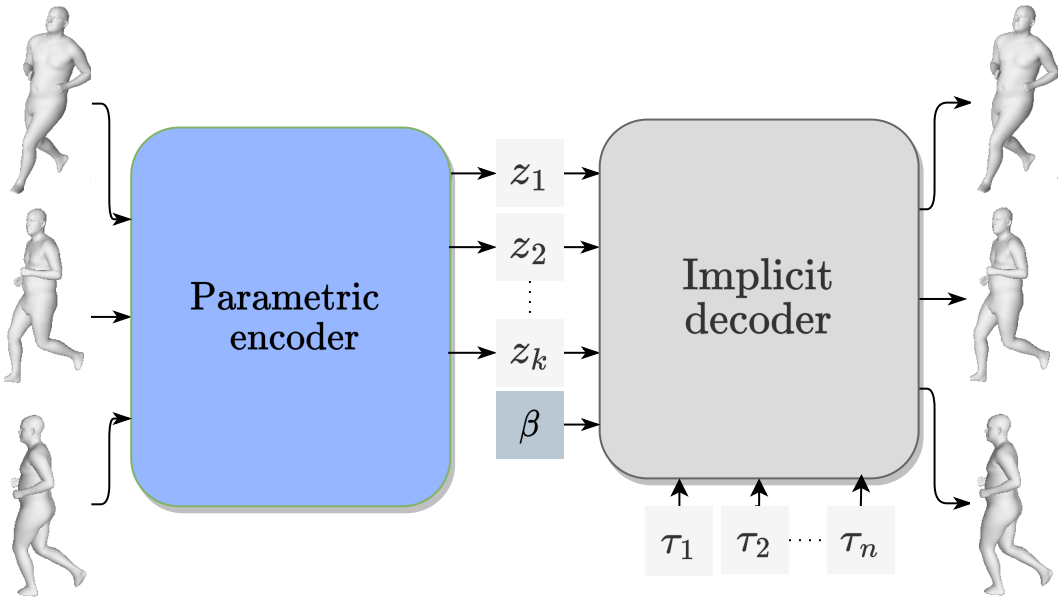


Figure 5.2: Method overview. Architecture consists of a seq2seq encoder (blue) that maps a human motion sequence into a sequence of latent primitives $z_1, \ldots, z_k$, and a temporally implicit decoder (grey block) that decodes the motion primitives $z_1, \ldots, z_k$, the latent morphology β and a series of timestamps $\tau_1, \ldots, \tau_n$ into a n-element sequence of parametric human body models.

Figure 5.2 provides a visual overview of our method. We use a data driven approach to learn our motion representation. To deal with the high dimensionality of the spatio-temporal data, we leverage a low-dimensional representation for each frame based on a human body model.

Given an input human motion sequence with variable duration and number of frames, the novelty of our approach is to encode it directly in a sequence of latent primitives, to better represent long motions. To address this sequence to sequence problem, we choose the transformer architecture [103]. This choice was guided by the transformer's capability to better retain long term correlations than its recurrent network counterparts.

Then we recover coherent output motions at arbitrary temporal resolution, allowing for more flexibility than existing motion representations. To do so, we

decode each latent primitive independently in a temporally implicit way, thereby converting the latent sequence to a sequence of parametric human body models for any given time instant. We then perform a weighted average of the sequence of parametric human body models using Gaussian temporal masks to recover the full motion. The Gaussian masks ensure smooth transitions between segments and allocate each segment to a contiguous temporal interval of motion. The temporally implicit nature of this decoder allows recovering the motion at any desired frame rate.

5.3 Method

This section provides details on the motion representation, a formal description of the sequential latent model, and an explanation of architecture choices.

5.3.1 4D sequence representation

We are interested in representing a large variety of motions performed by different subjects. To do so, we leverage the AMASS dataset [6], which is an aggregation of multiple motion capture data for which a fitted parametric body model is provided.

Similarly to the motion representation of Chapter 4 and other motion priors (*e.g.* [13, 64, 67, 117]), we follow the assumption that body shape remains constant over time , which allows to represent a motion sequence using a set of body shape parameters β, the joint rotations of a skeleton $\theta(\tau)$, and a 3D coordinate vector characterizing the displacement of the root joint $\gamma(\tau)$, where τ is the parameter controlling time. Our representation is agnostic to the parametric body model that is used. In our implementation, we use the SMPL body model [10] provided with the dataset. More data formatting details are provided in Sec. 5.4.1. Note that unlike in Chapter 4, we do not characterize motion as a discrete sequence of anchor frames but adopt a more flexible functional representation which is continuous in time.

We call $\chi(\tau) = \{\theta(\tau), \gamma(\tau)\}$ the motion parameter function and $m(\chi(\tau), \beta)$ the function that outputs a template aligned mesh corresponding to the parametric representation $\chi(\tau), \beta$. Our input motions are sequences $\{m(\chi(\tau_i), \beta), \tau_i\}_{i=1}^{n}$ consisting of n frames, where τ_i are the time stamps corresponding to the meshes. To simplify notation in the following Chapter, we refer to $\chi(\tau)$ as χ when considering the function object.

5.3.2 Latent representation

We represent the motion distribution $p(\chi)$ by two disentangled latent vectors, a sequential vector Z characterizing motion and a vector β characterizing body shape. This allows to formalize the motion distribution as $p(\chi) = \int\int p(\chi|Z, \beta)p(Z)p(\beta)dZd\beta$. While it is a similar high-level formalization was used in Chapter 4, we now model the motion prior $p(Z)$ as sequence of latent primitives, as outlined below, with the advantage of allowing for long-term sequences.

Morphology prior $p(\beta)$: As in Chapter 4, the body shape distribution $p(\beta)$ is given by a parametric body model. While our method is agnostic to the model used,

we use SMPL [10] in our implementation.

Motion prior $p(Z)$: Inspired by Ghorbani *et.al* [68], we subdivide motion into smaller segments in our representation. To achieve this, Z is represented as a sequence of independent latent primitives z_i such that $Z = \{z_i\}_{i=0}^{k}$. The primitive prior $p(z_i)$ is a normal distribution $p(Z) = \prod_{i=1}^{k} p(z_i) = \prod_{i=1}^{k} \mathcal{N}(0, I)$. This sequential representation in latent space allows for more flexibility in the representation than modeling the motion prior as a single latent vector, as done in existing works *e.g.* [13, 67, 117]. Intuitively, our representation subdivides a long motion sequence into shorter sequential actions, which allows for a compact representation.

5.3.3 Encoder

We use a transformer for our encoder network, and Fig. 5.3 shows its architecture. The transformer block of this encoder is similar to the original transformer [103] proposed for language translation.

The input of our encoder can be any temporal sampling of an input motion $\{\chi(\tau_i), \tau_i\}_{i=1}^{n}$, where the temporal resolution and the number of frames n may vary between input motions. Following the original implementation, the dimensionality of each frame $\chi(\tau_i)$ is adjusted through one perceptron layer. Note that the perceptrons applied to all frames share weights. Timestamps τ_i are subsequently concatenated to this representation, acting as positional encoding. The pairwise attention prediction of transformers is agnostic to sequence ordering. Positional encoding informs the model about the ordering. Using timestamps as positional encoding instead of the positional frame index proposed in [103] prevents the model from overfitting to the number of input frames.

The output of our encoder is a sequence of per primitive distributions $\{\mathcal{N}(\mu_i, \sigma_i)\}_{i=1}^{k}$ where each Gaussian approximates the distribution $p(z_i|\chi)$. The latent primitives are then sampled using Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ such that $z_i = \mu_i + \epsilon\sigma_i$.

5.3.4 Temporally implicit decoder

The decoder is implicit in time to allow for outputs at arbitrary frame rates. It has two objectives. First, ensuring that each primitive characterizes a contiguous motion segment. Second, combining motion segments, thereby generating a smooth motion without artifacts at transition points. Fig. 5.4 shows the architecture of the implicit decoder, which consists of two parts: primitive decoding and primitive combination.

Primitive decoding The latent primitives z_i are first decoded individually together with the body shape β to obtain a sequence of motion segments. A motion segment is characterized by its duration δ_i, a rigid transformation ρ_i and a parametric motion representation χ_i. In our architecture, a first MLP decoder outputs the duration $\delta_i = \mathcal{D}_{dur}(z_i, \beta)$. A second MLP outputs the rigid transformation $\rho_i = \mathcal{D}_{rigid}(z_i, \beta)$. Finally, a temporally implicit MLP characterizes the motion function χ_i as $\chi_i(\tau) = \mathcal{D}_{motion}(z_i, \tau - \Delta_i, \beta)$ where the temporal shift $\Delta_i = \sum_{j<i} \delta_j$ ensures the invariance of z_i *w.r.t* the starting time of the segment in the temporal reference frame of the global motion.

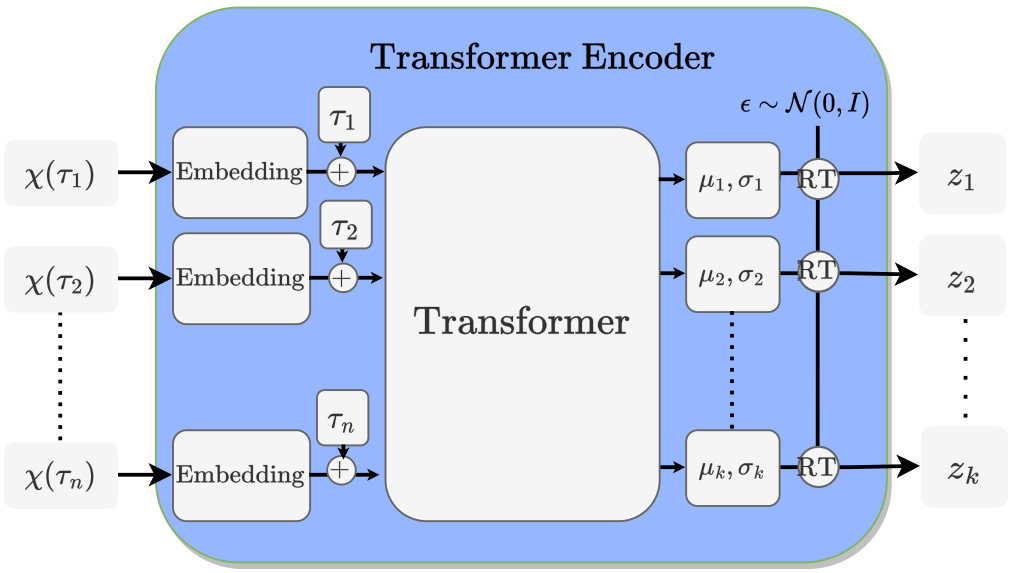


Figure 5.3: Encoder mapping an input sequence into a sequence of latent motion primitives $z_1, \ldots, z_k$. The embedding is a one layer perceptron. Time stamps τ_i are concatenated as positional encoding. Transformer outputs a sequence latent distributions μ_i, σ_i from which z_i are sampled using the reparametrization trick (RT).

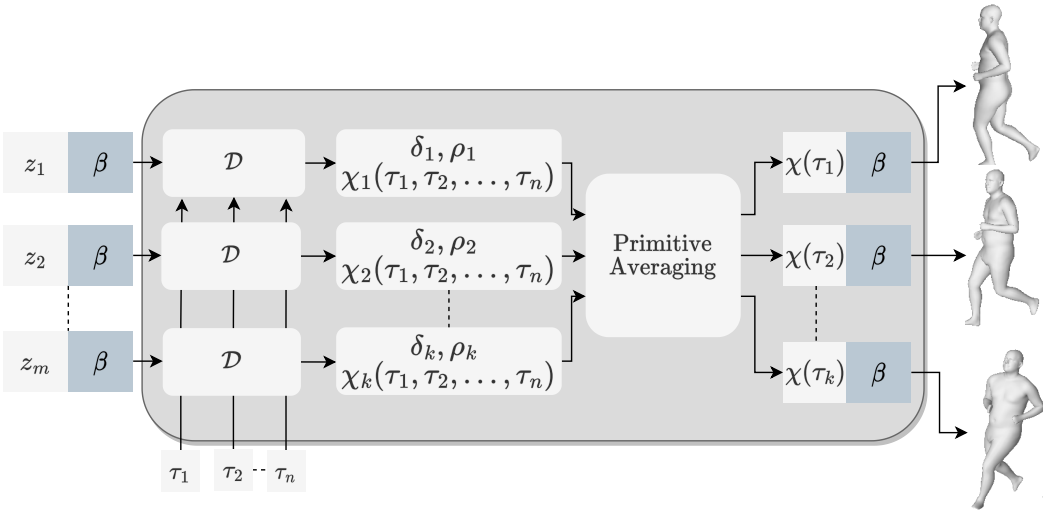


Figure 5.4: Implicit decoder. Given a sequence of k latent primitives z_i, a body shape β and n timestamps τ_j, the decoder outputs a sequence of body meshes parameterized by β, χ. The z_i are decoded independently into a motion segment characterized by its duration δ_i, a rigid transformation ρ_i, and a parametric motion representation χ_i which are subsequently combined to generate a dense 4D motion.

Primitive combination To combine the per-segment representations $(\delta_i, \rho_i, \chi_i)$ into a global motion representation χ, we perform a weighted average of $(\delta_i, \rho_i, \chi_i)$ using temporal masks. Each primitive is associated to a corresponding Gaussian mask $G_i(\tau) = e^{-\left(\frac{\tau - \frac{(\Delta_i + \delta_i)}{2}}{\delta_i/2}\right)^2}$ such that

$$\chi(\tau) = \frac{\sum_i G_i(\tau)(\rho_i * \chi_i(\tau))}{\sum_i G_i(\tau)}, \tag{5.1}$$

where $\rho_i * \chi_i(\tau)$ is the operation of applying the rigid transformation ρ_i to the body model parameters $\chi_i(\tau)$. This transformation consists of rotating the root joint for parameters $\theta_i(\tau)$, and rotating and translating the global displacements $\gamma_i(\tau)$.

The temporally implicit nature of $\mathcal{D}_{motion}$ alleviates the problem of averaging segments that may not be temporally aligned *w.r.t* a predefined frame rate. The Gaussian masks allocate each primitive to a contiguous temporal segment of the output motion and allow for smooth transitions between segments. The averaging of joint rotations is done in 6D representation space [118]. This leads to naturally combined results. For the masking function we also experimented with sigmoid masks instead of Gaussian masks. The equation of a sigmoid mask S_i is:

$$S_i(t) = Sigm(\lambda \frac{\tau - \Delta_i}{\delta_i}) - Sigm(\lambda(\frac{\tau - \Delta_i}{\delta_i} - 1)),$$

with $Sigm(x) = \frac{1}{1 + e^{-x}}$.

However, it was difficult to reach convergence with these masks due to high/vanishing gradients *w.r.t* parameters δ_i, Δ_i.

5.3.5 Training

The model is trained in a variational auto encoder (VAE) [52] setting with a reconstruction loss, and a Kullback–Leibler (KL) divergence loss between prior and posterior distributions. The total loss is

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_{KL}\mathcal{L}_{KL} \tag{5.2}$$

where

$$\mathcal{L}_{rec} = \mathcal{L}_{global} + \mathcal{L}_{segment}, \tag{5.3}$$

$$\mathcal{L}_{KL} = \frac{1}{k}\sum_{i=1}^{k} KL(\mathcal{N}(\mu_i, \sigma_i), \mathcal{N}(0, I)). \tag{5.4}$$

The reconstruction loss is divided into two terms. The first, $\mathcal{L}_{global}$, consideres global reconstruction between input and reconstructed output, including a per vertex distance to capture fine details and a distance in the parametric representation. It can be written as

$$\begin{aligned} \mathcal{L}_{global} = \frac{1}{n}\sum_{i=1}^{n} \Bigg(& \left|\left|\chi(\tau_i) - \chi_{GT}(\tau_i)\right|\right|^2 \\ + & \lambda_{3D}\left|\left|m(\tau_i) - m_{GT}(\tau_i)\right|\right|^2 \Bigg) \end{aligned} \tag{5.5}$$

with $||.||$ the L2-norm, $\chi_{GT} = \theta_{GT}, \gamma_{GT}$ the ground truth body model parameters and λ_{3D} a weighting coefficient that controls the relative influence of the per vertex distance. The second term $\mathcal{L}_{segment}$ acts as a per-segment reconstruction loss, which encourages segments to represent a plausible motion by optimizing

$$\mathcal{L}_{segment} = \frac{1}{kn} \sum_{i=1}^{n} \sum_{j=1}^{k} G_j(\tau_i) \Big|\Big| \rho_j * \chi_j(\tau_i) - \chi_{GT}(\tau_i) \Big|\Big|^2. \tag{5.6}$$

5.4 Evaluation

We start by outlining the implementation and data used to learn our representation. For evaluation, we study the representation's generalization to different frame rates and duration. We further study the influence of the sequential latent space and the segmentation learning on the representation. Furthermore, we compare our method to state-of-the-art methods for the application of spatio-temporal motion completion, both with and without correspondences.

5.4.1 Implementation and data

Implementation details

Our method is implemented using PyTorch and the Adam optimizer [119].

Unless stated otherwise, we set $m = 8$ and each latent vector has dimension $D = 256$. For the decoder $\mathcal{D}$, the two MLP $\mathcal{D}_{dur}$ and $\mathcal{D}_{rigid}$ are two layers MLP with ReLU activation after the first layer. $\mathcal{D}_{dur}$ has a sigmoid activation on the second layer to normalize the output duration between 0 and 1. The decoder $\mathcal{D}_{motion}$ is a three layer MLP with ReLU activations and LayerNormalization. When training the motion prior, we initially use a learning rate of 1e-4, which is reduced to 1e-5 after 20 epochs without improvement of the training loss, and further decreased to 1e-6 after 20 more epochs without improvement. This is done using the plateau scheduler of PyTorch. During the training phase we use $\lambda_{KL} = 0.0001$ and set $\lambda_{3D} = 0$ for the first 500 epochs because the 3D term slows down training significantly. Once we obtain good convergence after 500 epochs, we set $\lambda_{3D} = 1$ for 500 epochs. This significantly increases the pressure on trajectory reconstruction and gives a hierarchical importance to the joints, greatly reducing the reconstruction error in mm. We use a batch size of 16. The training phase takes between 1 and 2 days on a single GeForce RTX 2080Ti with 12 GB RAM.

Data Training is based on AMASS [6], which is a collection of different datasets parameterized by SMPL that contains a variety of motions and body shapes. Our training set consists of 9256 sequences. While our method is flexible *w.r.t* the number of input frames n, we fix $n = 100$ for training and train on motions of $3 - 5s$ by randomly sampling subsequences.

For testing, we consider two datasets: a test set based on AMASS and one based on multi-view reconstructions. The AMASS test set contains parts of AMASS coming from collections not used for training, thereby ensuring that all experiment are evaluated on unseen motions and unseen body shapes. We consider 136 sequences of at least $8s$ and test on sub-sequences of different duration starting at

the beginning of these sequences. We also consider the CHUM dataset introduced in Chapter 4. This prevents any eventual biases from having learned on AMASS. In this scenario, per-frame point clouds of roughly $10,000$ points were obtained by a multi-view stereo method [1] at a temporal resolution of 50 frames per second. As our model is now unrestricted in terms of motion variety, we consider 4s segments of all CHUM sequences including a complex cartwheel motion. In total, 170 motion sequences are used for testing. The 4s duration was chosen as it allows for comparison with other motion priors [64, 67, 117] that work without restriction on the motion types.

5.4.2 Generalization

To study the generalization of our representation to sequences of different duration and frame rates, we perform a forward pass on AMASS test sequences and consider the mean per joint position error (MPJPE) between input and output joints, which is a standard metric. This error is averaged over the sequence.

Generalization to different duration Our representation is learned with sequences of duration $3-5s$. We study its generalization to sequences of duration $0.2-8s$. To process sequences outside the training interval, we scale the timestamps to $[0,1]$. Fig. 5.5 shows the evolution of MPJPE for different sequence duration. Note that our model generalizes well to the simpler case of sequences with shorter duration than those used during training, and the error degrades gracefully for sequences of longer duration.

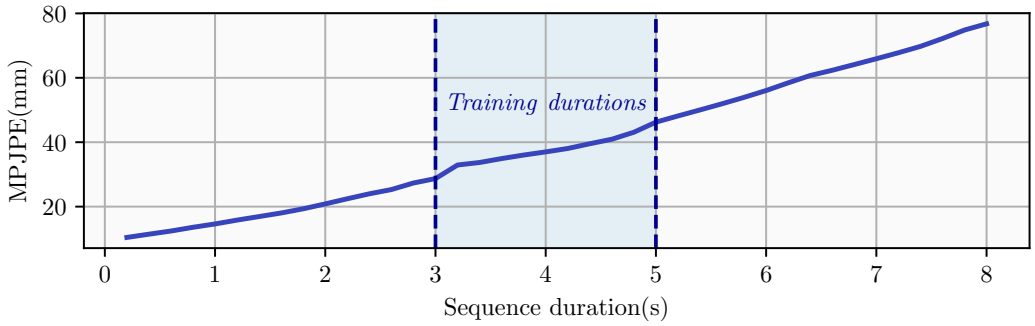


Figure 5.5: Generalization to sequence duration outside training duration ($3-5s$). Plot shows MPJPE (lower is better) for different sequence duration.

Generalization to different frame rate To study the influence of the frame rate on our representation, we re-sample all AMASS test sequences with frame rates between 5fps and 60fps before encoding and decoding them using our representation. Fig 5.6 shows the evolution of MPJPE for sequences sampled at different frame rates for 3 different sequence lengths. There is a slight drop in performance for frame rates below 10fps. When considering inputs with higher frame rates, the reconstruction error stays constant showing that our model generalizes to inputs with different frame rates.

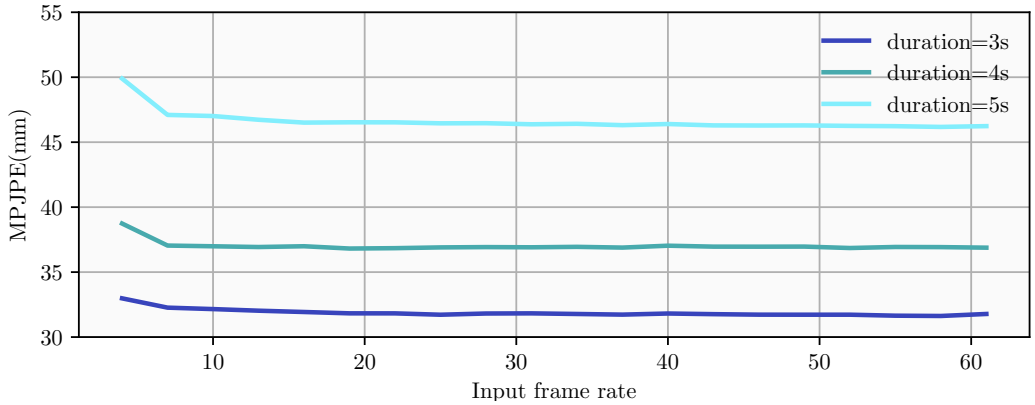


Figure 5.6: Generalization to different frame rates. Lines show MPJPE (lower is better) for different frame rates.

5.4.3 Influence of sequential representation

To evaluate the value of learning a sequence of latent primitives, we compare scenarios with varying numbers of primitives k for a fixed number of latent dimensions. Baseline $k = 1$ uses a single vector in latent space, as existing methods. We run this baseline with $D = 1024$, as increasing dimensionality in latent space further is computationally infeasible. Fig. 5.7 shows results for $k = 1, 2, 4$, where $k = 4$ uses the same number of latent dimensions, $D = 256$, as our selected representation. We consider MPJPE for sequences of increasing duration of AMASS test set. Using a sequence of latent primitives outperforms $k = 1$, and that using $k = 4$ outperforms $k = 2$, as expected. This shows that a sequential representation better generalizes to varying motions and duration.

5.4.4 Influence of segmentation learning

To evaluate the influence of learning segment duration, we compare our method to a baseline trained with fixed segmentation parameters $\delta_i = 1/k$ on the AMASS test set. As in the previous experiment, we set $k = 4, D = 256$ for both models and consider the generalization plot to sequences of different duration. Fig. 5.7 shows that the differences between the two models are minor. Allowing for flexible segments improves performance in the training interval while degrading gracefully for longer sequences.

5.4.5 Comparative evaluation

We show comparative evaluations *w.r.t* state of the art for spatio-temporal motion completion when input of fixed duration is degraded spatially and temporally.

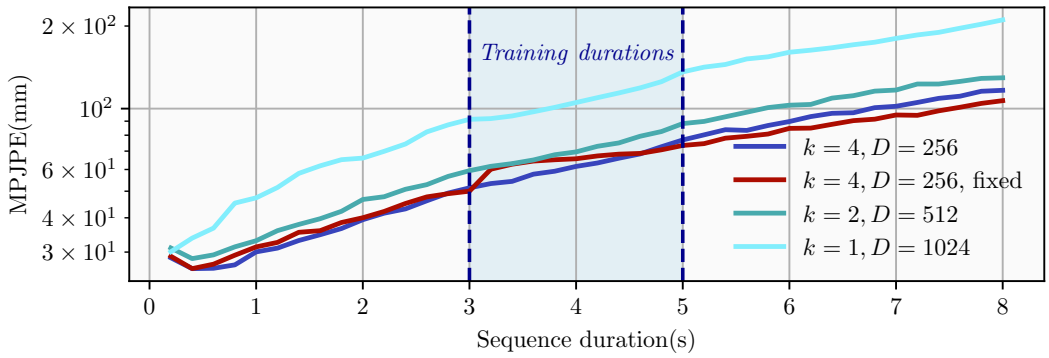


Figure 5.7: Value of using sequences of latent primitives and flexible segmentation. Lines show MPJPE (log scale, lower is better) for different sequence duration ($3-5s$ sequences used for training). Latent sequences with flexible segmentation $k = 4, D = 256$ perform best in training interval.

State-of-the-art methods

We compare to a strong baseline, two recent motion priors [67, 117][1], and a sequential representation [64].

Parametric baseline VPoser+SLERP The first baseline relies on the static pose prior VPoser [46], and outputs a latent pose representation per frame. As the global displacement is not encoded in VPoser, we additionally optimize per frame displacement. To increase the temporal resolution, we linearly interpolate between observed frames for displacement and use spherical linear interpolations (SLERP) for pose rotations. We call this baseline VPoser+SLERP.

Motion prior with frequency guidance [67] The first parametric prior uses frequency guidance and was trained for motions of fixed duration ($4s$) at 30 fps. This prior does not encode global displacements, so we optimize them per input frame and interpolate linearly for the remaining frames.

Hierarchical motion prior [117] The second parametric prior uses a hierarchical approach to encode motions of fixed duration ($2s$) at 30 fps.

Humor [64] The last baseline is a sequential representation that uses a sequence of frame transitions.

Motion completion without correspondences

Evaluation protocol

We evaluate the methods on the CHUM test set. The sequences are down sampled spatially to 100 and 1000 points per frame, and temporally to 5 and 10 fps to evaluate the robustness of the methods *w.r.t* degraded input signals. For each experiment, we reconstruct coherent 4D sequences at 30 fps, which is the proposed frame rate in existing methods [64, 67, 117]. We evaluate the error using a mean Chamfer distance over all frames of all test sequences. As the closest state-of-the-art methods consider sequences of fixed duration ($4s$ and $2s$, respectively), we perform our evaluation of sequences of duration $4s$, optimizing two latent vectors

[1]Note that [13] requires cyclic hip motion and is not applicable for the variety of motion considered in this experiment.

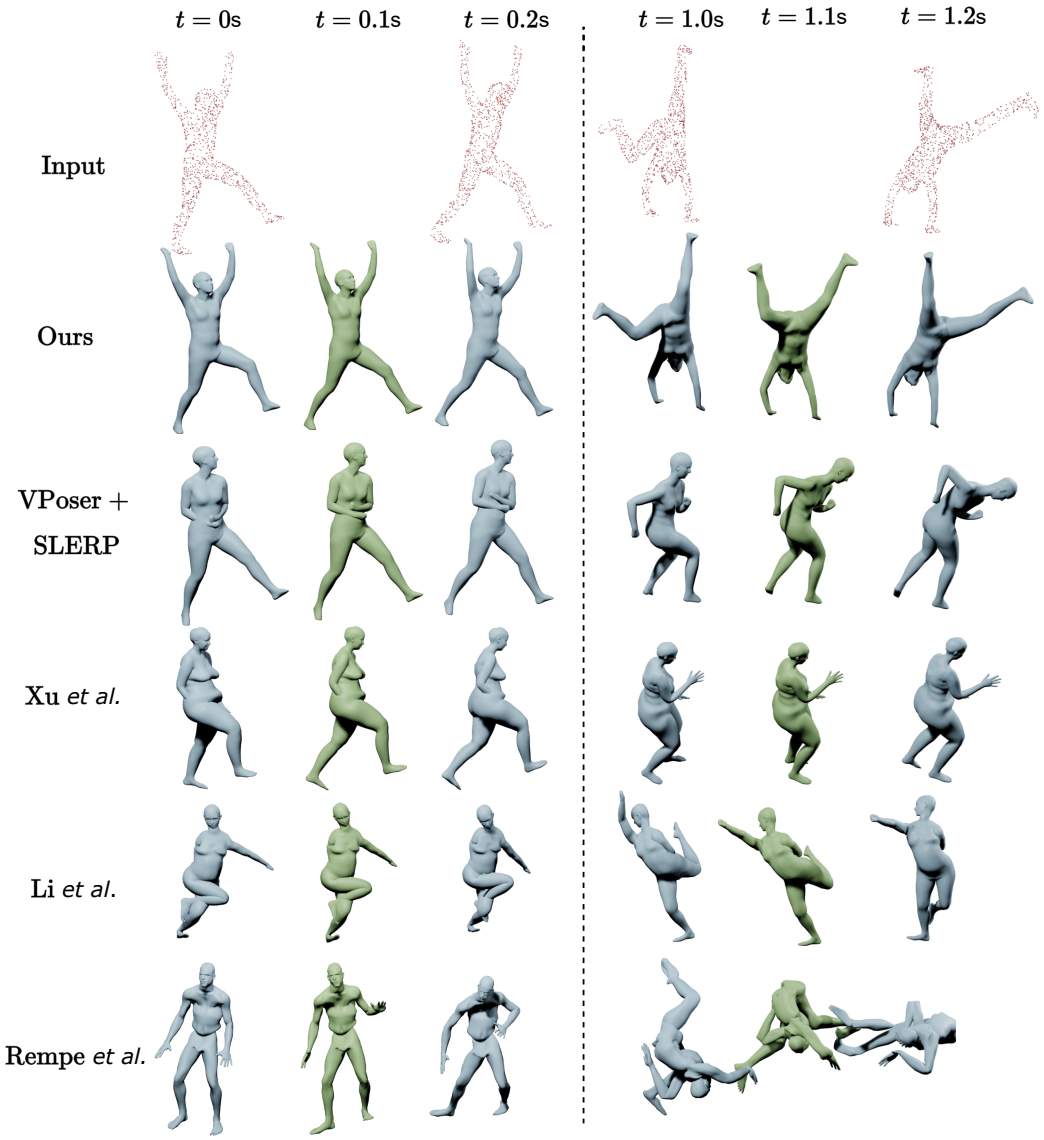


Figure 5.8: Comparison to state of the art on challenging example: a cartwheel with no temporal correspondences. We show frames close to the beginning and the end of the sequence. Our method estimates pose more precisely than other strategies. Blue meshes approximate input frames, green meshes are interpolated.

# Points per frame	100		1000		All	
Input fps	5	10	5	10	5	10
VPoser+SLERP	27	25	24	20	24	20
Xu *et.al* [67]	28	26	26	24	26	24
Li *et.al* [117]	83	79	52	48	42	40
Rempe *et.al* [64]	288	293	303	305	301	311
Ours	**21**	**17**	**17**	**13**	**17**	**13**

Table 5.1: Comparison to state-of-the-art average Chamfer distance (mm) (lower is better). Motion completion from different spatial (# points) and temporal (fps) resolutions.

for [117]. All comparisons are based on code and pre-trained models provided with the respective publications.

Latent optimization Given an observed sequence of sparse point clouds $\{P_i(\tau_i), \tau_i\}_{i=1}^n$ we optimize for each method a latent representation Z, β that best explains the observation. To initialize the latent representation, we adopt the initialization procedure proposed in the different papers. For Xu *et.al* [67], Li *et.al* [117] and VPoser+SLERP, we randomly sample from the prior $p(Z)$. For Rempe *et.al* [64], we use their initialization which relies on a per frame fitting of SMPL. For our method, Z is initialized with an encoder which is trained as a mapping function from sparse point cloud sequences to our sequential representation. For all methods, β is initialized to 0.

Then we solve for $\underset{Z,\beta}{argmin}(\mathcal{L}_{comp})$ with:

$$\mathcal{L}_{comp} = \mathcal{L}_{dense}(m(\mathcal{D}(Z, \beta, \tau_i), P_i(\tau_i)) + \lambda_{repr}\mathcal{L}_{repr}(Z), \tag{5.7}$$

where $\mathcal{L}_{dense}$ is the mean Chamfer distance between input point clouds and the vertices of the output meshes, $\mathcal{L}_{repr}$ is an additional regularization loss on the latent representation Z which was proposed in the respective papers. For Xu *et.al* [67], Li *et.al* [117] and VPoser+SLERP, it constrains the latent motion Z to stay close to the origin. For Rempe *et.al* [64], it both constrains the latent motion and the shape to stay close to a given prior and in our method, it constrains the latent motion Z to stay close to its initialization. We set $\lambda_{repr} = 0.01$ for all methods.

Results Table 5.1 reports results for input increasingly sparsely sampled in space and time. Our method outperforms baselines by a large margin, especially when considering very sparsely sampled input, and degrades gracefully for decreasing input resolutions.

Fig. 5.8 shows qualitative results on a challenging cartwheel sequence of CHUM. While our method generates a motion close to the input point clouds, all other methods fail to capture the global orientation of the motion because they do not encode global displacement in their latent representation (VPoser+SLERP, Xu *et.al*), or because the global displacement is estimated erroneously (Li *et.al*, Rempe *et.al*). While the method of Rempe *et.al*showed great results on completion tasks with correspondences, its optimization fails when considering unstructured

data, because it relies on a per frame initialization that requires correspondences. In contrast, our method is robust to global orientation and translation. As it was trained to predict rigid transformations and encodes global displacement, it does not suffer from discontinuities at segment transitions. It also learned detailed motion features thanks to the small temporal windows covered per segment, which is visible on the cartwheel motion where our model correctly captured arm positions.

5.5 Conclusion and future works

This chapter presented a temporally implicit spatio-temporal representation of motion using a sequence of latent primitives. Using a sequence of latent primitives characterizing temporal segments of motion allows for a gain in precision which is more efficient than increasing the dimensionality of a single latent space representation similar to the one introduced in Chapter 4. Another key advantage of this representation is its added flexibility *w.r.t* sequence duration and frame rate made possible by the implicit representation of the temporal dimension. This method extended the representation power of Chapter 4 to a variety of human motions and was demonstrated experimentally to outperforms state-of-the-art motion priors on a completion task from sparsely sampled point clouds.

While we demonstrated that our representation effectively encodes long-term locomotion information accurately, it cannot represent the dynamics of fine-scale geometric details. Enhancing this representation to allow for fine-scale dynamics has the potential to be applicable to dressed humans with accessories and would be interesting. Future work could also investigate sequential learning which takes into account dependencies between motion primitives. It could prove useful for long term motion synthesis and be promising to generate a coherent sequence of primitives by prior sampling.

6

Correspondence-free online human motion retargeting

Résumé

Dans ce chapitre, nous proposons d'étudier une autre approche générative plus spécialisée qui s'attaque au problème du reciblage. Le reciblage du mouvement humain est le processus d'animation d'un personnage cible avec la séquence de mouvement d'un personnage source.

Nous étudions ici ce problème dans le cas difficile du reciblage sans correspondance qui prend en entrée des données 4D non structurées pour lesquelles on ne dispose ni de correspondances entre la forme de la source et celle de la cible, ni de correspondances entre les différentes images du mouvement de la source. Cela permet d'animer directement une forme humaine cible à l'aide de séquences arbitraires de personnes en mouvement, éventuellement capturées à l'aide de plateformes d'acquisition 4D.

Nous y parvenons en combinant les avantages du reciblage du mouvement squelettique et du reciblage du mouvement basé sur la surface afin de connaître à la fois le contexte temporel et les détails géométriques. Plus précisément, nous combinons le reciblage des mouvements squelettiques et le skinning automatique, ce qui permet de traiter des données spatio-temporelles de haute dimension.

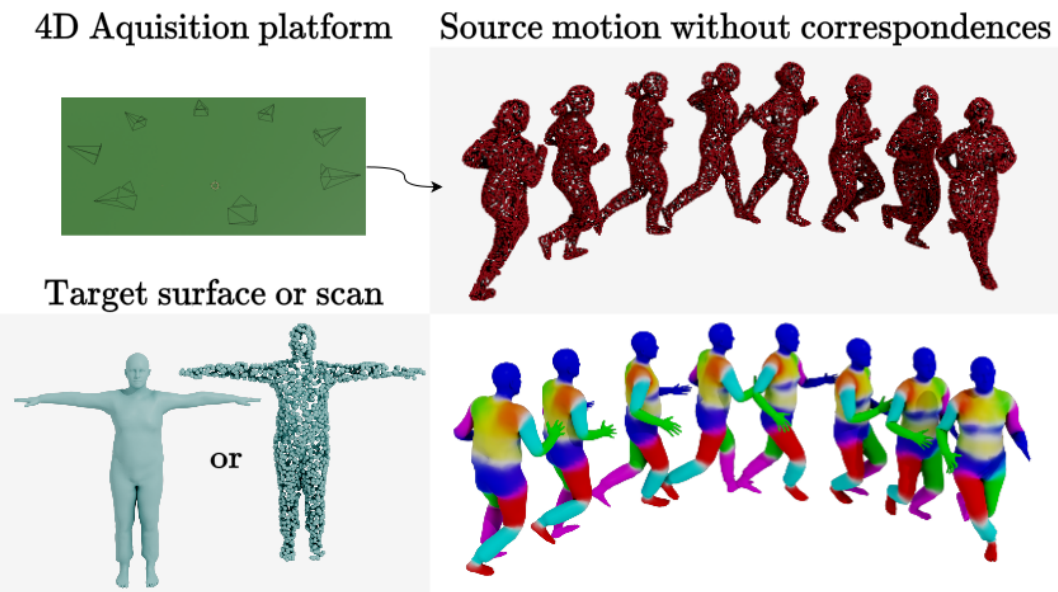


Figure 6.1: Given an untracked source motion (top) and a target body shape (bottom left), our method animates the target with the source motion, preserving temporal correspondences of the output motion (bottom right).

6.1 Introduction

Chapter 4 and 5 proposed generic motion priors to represent dense human motion in a low dimensional space. In this chapter, we propose to investigate another more specialized generative approach that tackles the retargeting problem. Human motion retargeting is the process of animating a target character with the motion sequence of a source character.

Applications of motion retargeting include video gaming, movie making, avatar animation, and augmenting existing datasets of 4D body motions *e.g.* [6, 91, 120, 121]. Here, we study this problem in the difficult case of correspondence-free retargeting that takes as input unstructured 4D data for which neither correspondences between the source and target shape nor correspondences between different frames of the source motion are given. This allows to directly animate a human target shape with arbitrary sequences of humans in motion, possibly captured using 4D acquisition platforms. Fig. 6.1 illustrates the problem's input and output.

Motion retargeting has recently made significant progress and leads to impressive results. Skeleton-based methods allow for retargeting while taking long-term temporal context of about 2 seconds into account *e.g.* [70, 77, 122]. Surface deformation-based methods allow retargeting geometric details while possibly capturing short time dynamics of motion *e.g.* [35, 39, 40, 89].

Most of these retargeting works take as input structured data in the form of skeletons or template-aligned surfaces for which correspondence information is known. Solving the retargeting problem for unstructured 3D data is challenging as computing correspondences is a combinatorial problem.

Recently, a first solution for correspondence-free motion retargeting has been proposed [86]. This work introduces a generic motion prior to describe how body

shapes move in a low dimensional space, and takes long-term temporal context into account. Unlike the motion priors proposed in Chapters 4 and 5, this concurrent prior has the advantage of operating with correspondence-less data. While this opens the door to novel solutions, during inference the method is limited to retarget sequences of fixed length.

In this work, we propose the first correspondence-free method for online motion retargeting that learns long-term temporal context. Especially, we are able to learn context from $1s$ of motion sequences, which improves the accuracy of motion retargeting even with challenging examples. We achieve this by combining the advantages of skeletal motion retargeting and surface-based motion retargeting to learn about both temporal context and geometric detail. More specifically, we combine skeletal motion retargeting and automatic skinning, thereby allowing to handle high-dimensional spatio-temporal data. The reason is that skeletal retargeting methods achieve good results by encoding long-term temporal context [77] and that recently, methods on automatic skinning have been proposed that we can include in our inference model [26, 123–125]. Our hypothesis is that the locomotion information we want to extract from the source sequence is explained at a skeletal level, while the dense surface details are intrinsic to the target shape.

In practice, our method is divided in three modules. First, a skeleton regressor allows to transition between unstructured 3D point clouds and skeletal representations. Second, a skeleton-based motion retargeting method allows to transfer the source motion to the target skeleton. Third, an automatic skinning predictor, which combined with classical linear blend skinning (LBS) reposes the unstructured target point cloud.

Our main contributions are summarized below:

- We propose the first correspondence-free online approach for dense human motion retargeting that learns temporal context.
- We demonstrate that long-term temporal context improves the accuracy of motion retargeting.
- We demonstrate state-of-the-art results for correspondence-free methods in both geometric detail preservation and skeletal-level motion retargeting.

This project was made in equal collaboration with Rim Rekik Dit Nekhili, another Ph.D. student of Morpheo team. Technically, she focused on the skeletal regression module while I implemented the skeletal retargeting module and the skinning predictor. Our code is available at `https://gitlab.inria.fr/rrekikdi/human-motion-retargeting2023`

6.1.1 Positioning

Table 6.1 summarizes the positioning of our work *w.r.t* state-of-the-art. We classify approaches based on four criteria: using long-term temporal context ($0.5s$ or more) for training, allowing for online inference, modeling geometric detail, and operating on unstructured data for which no correspondences are known. By combining the advantages of skeleton-based retargeting and skinning priors, we propose the first

correspondence-free retargeting approach that learns long-term temporal context, allows for arbitrary duration at inference, all while modeling geometric detail at the surface level.

Method	Temporal context	Online inference	Geometric detail	Unstructured
Skeleton-based				
[77, 78, 122, 126]	✓	✓	✗	✗
[87]	✓	✓	✓	✗
Shape deformation transfer				
[40, 89, 127]	✗	✓	✓	✗
[39, 88]	✗	✓	✓	✓
Motion priors				
[13]	✓	✗	✓	✗
[86]	✓	✗	✓	✓
Ours	✓	✓	✓	✓

Table 6.1: Positioning w.r.t state-of-the-art retargeting approaches. We propose the first correspondence-free retargeting approach that learns long-term temporal context, allows for arbitrary duration at inference, all while modeling geometric detail at the surface level.

6.2 Motion retargeting model

In this section, we introduce notations, give an overview of our retargeting model and detail for all its stages. We then discuss our implementation and training strategy.

6.2.1 Overview

The input source motion is characterized by a sequence of n point clouds $\{\mathcal{S}_i^A\}_{i=1}^n$ without temporal correspondences and the target shape B by a single point cloud, possibly with connectivity information, with B in T-Pose $\mathcal{S}_{tpose}^B$. Our objective is to generate the sequence of retargeted scans $\{\mathcal{S}_i^B\}_{i=1}^n$, where $\mathcal{S}_i^B$ imitates the pose of $\mathcal{S}_i^A$ while retaining the body shape of $\mathcal{S}_{tpose}^B$, and is in correspondence with $\mathcal{S}_{tpose}^B$. Fig. 6.2 gives a visual overview of our method.

A common strategy for motion retargeting is to disentangle the high-level motion from the shape deformation caused by the body shape, and to combine the source motion with the target body shape. This is computationally expensive when considering densely sampled input surfaces containing thousands of vertices per frame, especially if the surfaces are not in correspondence, which hinders existing approaches to learn long-term temporal context.

We overcome this challenge by making the hypothesis that source motion information is fully explained at a skeletal level, while geometric detail information is encoded in the target. This hypothesis leads to a three-step framework.

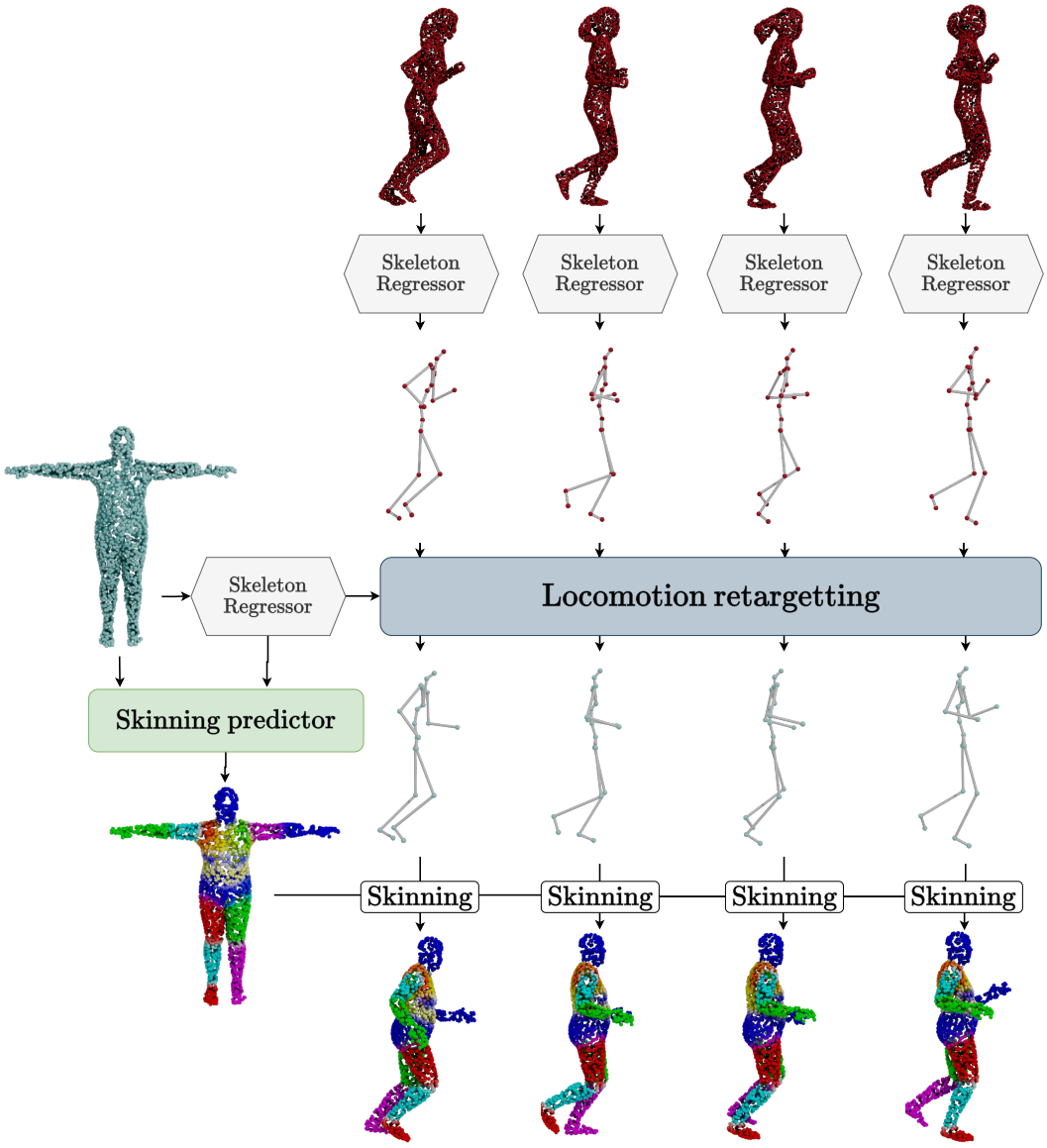


Figure 6.2: Our method takes a source sequence of unstructured point clouds along with a target point cloud as input and outputs the target character performing the input motion. The method proceeds in three stages: the first one (gray boxes) extracts per-frame skeletal representations from the input source sequence, the second one (blue box) retargets the locomotion to the target character at the skeleton level, and the third one (green box) adds the surface details of the target character to the resulting motion using densely predicted skinning parameters.

First, we extract the motion information from the source motion by predicting a sequence of skeletons $\{\mathcal{J}_i^A\}_{i=1}^n$ from $\{\mathcal{S}_i^A\}_{i=1}^n$. This prediction is performed independently per frame to allow processing long sequences. To obtain a skeletal representation from correspondence-less point clouds, we regress a set of joint positions with a Skeleton Regressor module (SKR). This skeletal parameterization has three major advantages. It provides correspondence information between input frames, drastically reduces the dimensionality of the data and fully encodes the shape induced variability in bone length.

Second, from the resulting skeletal parameterization of the source $\{\mathcal{J}_i^A\}_{i=1}^n$, we extract high level motion information and retarget this motion to the skeletal parameterization of the target $\mathcal{J}_{tpose}^B$. Working on a skeletal representation allows training using long-term context without running into complexity issues. To do so, we build upon a context-aware skeletal motion retargeting model (SMRM) [77] which given $\{\mathcal{J}_i^A\}_{i=1}^n$ and $\mathcal{J}_{tpose}^B$ outputs the relative joint rotations $\{\theta_i^B\}_{i=1}^n$.

Third, we recover shape details of B to compute the final animation. We achieve this by animating $\mathcal{S}_{tpose}^B$ using $\{\theta_i^B\}_{i=1}^n$ via skinning. As $\mathcal{S}_{tpose}^B$ is sampled arbitrarily, we learn a dense skinning prior (SKIN) as an implicit function that can be evaluated at any position on the body surface. More specifically, SKIN is composed of two spatially implicit networks: one that predicts per joint skinning weights for each point in B, and one that predicts pose-corrective offset for each point in B. Using LBS, the predicted skinning weights and the input rotations θ^B allow animating $\mathcal{S}_{tpose}^B$ one frame at a time. The pose-corrective offsets are added to the points of $\mathcal{S}_{tpose}^B$ to attenuate skinning artifacts.

A key advantage of our framework is its low computational complexity for increasing temporal context during training. Another advantage of our method is that the model can generalize to several sources of variability. It can generalize to various source shapes thanks to SKR, to various source motions thanks to SMRM and to various target shapes thanks to SKIN. This flexibility allows us to retarget the raw untracked output of a multi-view platform to virtual avatars or clothed body shapes using a model trained on synthetic naked body shapes. The following provides details about each of the parts.

6.2.2 Skeleton Regressor

In the first step, the skeleton regressor SKR extracts skeletal motion from the sequence of point clouds $\mathcal{S}_1^A, \mathcal{S}_2^A, \ldots, \mathcal{S}_n^A$ by treating each point cloud $\mathcal{S}_i^A$ independently, as well as from point cloud $\mathcal{S}_{tpose}^B$. The input scans are centered by subtracting their centroid and scaled in the unit cube. The skeleton is parameterized by 22 joint positions corresponding to joints of the SMPL body model [10].

The main challenge of this module comes from the unstructured point cloud input. To operate on unstructured point clouds, SKR should be robust *w.r.t* the number and order of the observed points. To achieve this, we use the PointFormer [128] architecture to extract order-invariant features from the point cloud, followed by a multi layer perceptron to regress the joint positions from these features. We choose the PointFormer architecture as it considers local features of the point cloud allowing for precise joint predictions and good generalization to unseen poses. This

step allows recovering parameterized skeletons as

$$\mathcal{J} = \text{SKR}(\mathcal{S}). \tag{6.1}$$

6.2.3 Skeletal motion retargeting

The second module retargets $\{\mathcal{J}_i^A\}_{i=1}^n$ to $\mathcal{J}_{tpose}^B$. This retargeting to a character of different proportions is challenging because of the noise introduced by SKR and the lack of ground truth pairings between a motion and its retargeted version, which implies that the training needs to be unsupervised.

Unsupervised skeleton retargeting has been successfully studied in prior works, and we base our method on SMRM [77]. SMRM starts by extracting high level locomotion features from $\{\mathcal{J}_i^A\}_{i=1}^n$ using a first recurrent network. These features are leveraged along with $\mathcal{J}_{tpose}^B$ to generate joint rotations $\{\theta_i^B\}_{i=1}^n$ using a second recurrent unit. These rotations are then applied to $\mathcal{J}_{tpose}^B$ using a differentiable forward kinematics layer to generate the retargeted skeletal motion $\{\mathcal{J}_i^B\}_{i=1}^n$. A cycle consistency objective is used to overcome the lack of direct supervision. First, $\{\mathcal{J}_i^A\}_{i=1}^n$ is retargeted to $\{\mathcal{J}_i^B\}_{i=1}^n$ and then $\{\mathcal{J}_i^B\}_{i=1}^n$ is retargeted back to $\{\mathcal{J}_i^A\}_{i=1}^n$. This cycle consistency is combined with adversarial training to encourage realism of $\{\mathcal{J}_i^B\}_{i=1}^n$.

We make two changes to SMRM that significantly improve retargeting results in our scenario. First, rather than representing joint rotations using quaternions, we represent them in 6D as this representation was shown to be beneficial for deep learning applications [118]. Second, we add a cycle consistency loss on the joint rotations $\{\theta_i^B\}_{i=1}^n$, as explained in Sec. 6.2.5. To simplify notation, we call our improved model SMRM in the following. This step allows to retarget a skeletal motion as

$$\{\theta_i^B\}_{i=1}^n = \text{SMRM}(\{\mathcal{J}_i^A\}_{i=1}^n, \mathcal{J}_{tpose}^B). \tag{6.2}$$

6.2.4 Skinning predictor

The third stage takes as input $\mathcal{S}_{tpose}^B$, $\mathcal{J}_{tpose}^B$, and $\{\theta_i^B\}_{i=1}^n$, and outputs an animation $\{\mathcal{S}_i^B\}_{i=1}^n$ of body shape B performing the source motion. The main challenge come from the arbitrarily sampled $\mathcal{S}_{tpose}^B$ and from input noise introduced by the previous stages.

A common strategy to animate 3D meshes given their skeletal motion is to associate surface vertices to the skeleton joints by a set of skinning weights and to animate the shapes using skinning techniques. While recent works in this area show impressive results [26, 124, 125, 129, 130], they are typically applicable to registered data or supervised with skinning weights during training, which are not given in our case. To allow for arbitrarily sampled input point clouds, we learn an implicit skinning prior SKIN that can be evaluated for any surface point of a human in T-pose. Inspired by works that build human shape spaces [10, 131], we model SKIN using two parts: a skinning weights predictor and a pose-corrective offset predictor. The pose corrective offset is added to vertices in T-pose before applying LBS in order to diminish the artifacts caused by LBS. We choose this model for its simplicity and for its excellent performance when modeling human deformations.

The skinning weights predictor network W_{net} takes the target joints $\mathcal{J}^B_{tpose}$ and a 3D point p of $\mathcal{S}^B_{tpose}$ as input and outputs one skinning weight per joint $W = W_{net}(\mathcal{J}^B_{tpose}, p)$. Inspired by [130], we constrain the predicted skinning weights to sum to 1 using a softmax activation. The pose-corrective offsets network O_{net} takes relative joint rotations θ^B and p as input and outputs an offset vector in $\mathbb{R}^3$ as $o^B = O_{net}(\theta^B, p)$. Given the target joints $\mathcal{J}^B_{tpose}$, the set of rotations $\{\theta^B_i\}^n_{i=1}$, the skinning weights W^B_j and the pose corrective offsets $\{o^B_{i,j}\}^n_{i=1}$ of point p_j of $\mathcal{S}^B_{tpose}$, we use LBS to generate the j-th vertex of the retargeted sequence as

$$\{\mathcal{S}^B_{i,j}\}^n_{i=1} = \text{LBS}(\{\theta^B_i\}^n_{i=1}, \mathcal{J}^B_{tpose}, W^B_j, \{p_j + o^B_{i,j}\}^n_{i=1}). \tag{6.3}$$

The skinning predictor is modeled with two three layer MLPs. The first MLP models W_{net} and is followed by a softmax activation. The second MLP models O_{net}.

6.2.5 Training

Inspired by recent works on related problems [86,132–134], we choose a stage-wise training strategy to improve training stability and reduce computational complexity.

Data To train all three parts, we use the AMASS dataset [6], which contains a collection of motion capture datasets that have been fitted by the parametric body model SMPL [10] to obtain dense per-frame representations. For each frame, aligned surface and skeleton information is given. As training data, we consider a subset of 120 body shapes, seen performing 2536 different motions for a total of 65000 seconds of motion. As validation data, we consider a subset of body shapes seen performing 24 different motions for a total of 147 seconds of motion and as testing set we consider body shapes seen performing 44 motions for a total of 1715 second of motion, we refer to this test set as AMASS test set in the following.

All the motion sequences are temporally aligned at 30 frames per second (FPS). To train SKR and SKIN, we sample one frame for each second of motion.

Skeleton regressor We train the skeleton regressor using static 3D meshes provided by AMASS with their corresponding skeletons. To avoid learning the topology bias of the SMPL template mesh, which provides correspondence information, we randomly uniformly sample N points on the surface of each 3D mesh and add Gaussian noise to generate input scans $\mathcal{S}$. During training, we minimize as loss the mean squared error (MSE) between the ground truth SMPL joints $\mathcal{J}$ and the predicted joints:

$$\mathcal{L}_{\text{SKR}} = MSE(\text{SKR}(\mathcal{S}), \mathcal{J}). \tag{6.4}$$

Skeletal motion retargeting To train this part, we randomly sample sequences of fixed duration from AMASS and consider their skeletal motion only.

The loss function used during training includes the loss functions used in the baseline method [77], which are a cycle consistency loss and an adversarial loss. The method retargets $\{\mathcal{J}^A_i\}^n_{i=1}$ to $\mathcal{J}^B_{tpose}$, leading $\{\theta^B_i, \mathcal{J}^B_i\}^n_{i=1}$, which is then retargeted to $\mathcal{J}^A_{tpose}$ to lead $\{\hat{\theta}^A_i, \hat{\mathcal{J}}^A_i\}^n_{i=1}$. The cycle consistency loss is defined as

$\mathcal{L}_{cycon} = MSE(\hat{\mathcal{J}}_i^A, \mathcal{J}_i^A)$. The adversarial loss leverages a discriminator network which is trained in a min-max game with the retargeting network to differentiate between real and retargeted motions and is denoted by $\mathcal{L}_{adv}$.

To improve performance, we add two additional loss functions to regularize the generated motions. Our first loss is a cycle consistency loss on the joint rotations, which allows preventing unrealistic motions, and can be written as $\mathcal{L}_{rot} = MSE(\hat{\theta}_i^A, \theta_i^A)$. The second loss allows for temporal smoothing of the motions by aiming to reconstruct velocities, and can be written as $\mathcal{L}_{smooth} = MSE(\Delta\hat{\mathcal{J}}_i^A, \Delta\mathcal{J}_i^A)$, with $\Delta\mathcal{J}_i = \mathcal{J}_{i+1} - \mathcal{J}_i$. The complete loss for the retargeting module is

$$\begin{aligned} \mathcal{L}_{\text{SMRM}} &= \mathcal{L}_{cycon} + \mathcal{L}_{adv} \\ &+ \lambda_{rot}\mathcal{L}_{rot} + \lambda_{smooth}\mathcal{L}_{smooth} \end{aligned} \tag{6.5}$$

with $\lambda_{rot} = 0.01$ and $\lambda_{smooth} = 0.1$.

Skinning predictor The skinning predictor is trained using static AMASS data by considering pairs of frames $\{\mathcal{S}_{tpose}, \mathcal{S}\}$ of a same person in correspondence and with known rotations θ that explain the pose of $\mathcal{S}$. To preserve correspondences while removing the bias due to SMPL topology, the models are uniformly resampled while preserving correspondence information.

To train the skinning predictor network, we consider point p of $\mathcal{S}_{tpose}$ and its corresponding point p' in $\mathcal{S}$, and minimize the reconstruction loss after deforming $\mathcal{S}_{tpose}$ to pose θ using LBS as:

$$\mathcal{L}_{\text{SKIN}} = \sum_{(p,p')\in(\mathcal{S}_{tpose},\mathcal{S})} ||p' - LBS(\theta, \mathcal{J}_{tpose}, W, p + o)||^2, \tag{6.6}$$

where $W = W_{net}(\mathcal{J}_{tpose}, p)$ and $o = O_{net}(\theta, p)$.

6.3 Experiments

We now evaluate our main contributions: that learning from long-term temporal context improves the results, that our method outperforms state-of-the-art, and generalizes to a large variety of target shapes and source motions.

We first show quantitatively that considering long-term temporal context improves the accuracy of motion retargeting. Second, we present quantitative comparisons and comparisons via user studies to state-of-the-art results for both geometric detail preservation and skeletal-level motion retargeting on challenging shape transfers with both naked and clothed target shapes where both shape and motions are unseen during training. Finally, we show results that retarget the raw output of a 4D multi-view acquisition platform to new target characters.

Data For a fair evaluation of the different methods considered in our comparison, we evaluate all methods on two test sets. First, a test set of representative naked human body shapes performing the same set of varying long-term motions. To build this dataset, we consider 4 body shapes created using the SMPL model [10], as is commonly done when evaluating human deformation transfer methods *e.g.* [89]. We

sample body shapes at ± 2 standard deviations along the first 2 principal components to cover the main sources of variability of human body shape. Skeleton-based retargeting methods *e.g.* [87] commonly evaluate on the Mixamo dataset [1]. Inspired by this, we create and retarget a set of 4 motions to all body shapes using Mixamo to generate corresponding ground truth motions. We call this SMPL test set in the following.

The second test set considers characters with clothing performing long-term motions. This test set allows to evaluate the generalization of different methods to geometric detail in the target shape. To generate this test set, we take 4 characters with clothes from CAPE [135] and retarget 3 motions from the Mixamo dataset to each of these models. We call this CAPE test set in the following.

Note that for both of SMPL and CAPE test sets, none of the body shapes or motions were observed by any of the methods during training. Furthermore, by using Mixamo to retarget the same motion to different body shapes, ground truth retargeting results are available for quantitative testing.

We also propose an additional test set which considers raw untracked data acquired using a multi-view camera setup [13]. For this dataset, no ground truth retargeting is available, and we provide qualitative results for our method. We call this multi-view test set in the following.

Evaluation metrics The goal is to evaluate the retargeting results in terms of the overall preservation of the motion and the detail-preservation of the target geometry.

To evaluate the overall motion, we use two complementary metrics that operate exclusively on the skeletal level. First, we consider the mean-per-joint error (MPJPE) between the ground truth and the retargeting result, which evaluates the overall accuracy of the joint positions, and the Procrustes aligned MPJPE (PA-MPJPE), which eliminates the error in global displacement. Second, to evaluate motion smoothness, we consider the mean acceleration difference between ground truth predicted motions (Acc) and its Procrustes aligned (PA-Acc) version.

To evaluate detail-preservation of the target geometry, we use two complementary metrics that operate on the surface. The first are mean-per-vertex distance (MPVD) and Procrustes aligned MPVD (PA-MPVD) between the ground truth and the retargeting result, which evaluate the global extrinsic accuracy of the predicted surface. Second, to evaluate the preservation of intrinsic geometry, we compute a mean difference in edge length (MDEL) between the ground truth and the retargeting result. As we operate on point clouds, we create edges by connecting the 6 closest neighbors of every point in the ground truth.

6.3.1 Learning with long-term temporal context

Our first experiment demonstrates that considering temporal context beyond a few frames during training is beneficial to motion retargeting. We train our model with motion sequences containing different numbers of frames, *i.e.*for each model, all training sequences have a fixed number of frames, which ranges from 5 frames (similar to shape deformation transfer methods [88, 89]) to 60 frames (similar to

[1] https://www.mixamo.com

	Skeletal motion				Detail preserv.		
	MPJPE (m) ↓	PA-MPJPE (m) ↓	Acc ↓	PA-Acc ↓	MPVD (m) ↓	PA-MPVD (m) ↓	MDEL (mm) ↓
Naked target shapes from SMPL test set							
H4D [86]	0.238	0.096	**0.019**	0.014	0.152	0.078	402.238
NPT [39]	0.388	0.165	0.024	0.014	0.227	0.132	2.739
Ours	**0.158**	**0.043**	0.021	**0.008**	**0.134**	**0.040**	**0.524**
Clothed target shapes from CAPE test set							
H4D [86]	0.161	0.091	**0.019**	0.014	0.096	0.074	395.683
NPT [39]	0.373	0.168	0.022	0.013	0.173	0.135	2.845
Ours	**0.105**	**0.040**	0.023	**0.008**	**0.075**	**0.038**	**0.539**

Table 6.2: Comparison to state-of-the-art on naked (top) and clothed (bottom) target shapes. Best performing scores shown in bold.

skeleton based methods [77, 122, 126]). Table 6.3 shows the results. Including long-term context improves almost all metrics up to 30 frames. In all following, we use the model trained with sequences of 30 frames.

6.3.2 Quantitative comparison to state-of-the-art

We now present a comparative analysis to state-of-the-art motion retargeting methods.

Competing methods As summarized in Table 6.1, there are three lines of existing methods. Skeleton-based retargeting methods are not comparable to our approach as they require handcrafted skinning weights as input, which are not available for our test sets. We therefore compare our method to state-of-the-art correspondence-free deformation transfer methods and motion priors. For deformation transfer methods, we compare to NPT [39]. Aniformers [88] requires ground truth pairings of different individuals performing the same motion during training, which are not available in our case, making comparison impossible. For motion priors, we compare to H4D [86]. We provide a full identity sequence to this method to extract body shape parameters.

Quantitative results Table 6.2 provides quantitative results when considering naked and clothed body shapes of the SMPL and CAPE test sets, respectively. Our method significantly outperforms NPT on almost all metrics on both datasets. The reason is that NPT operates per frame without any temporal context and sometimes leads to results that are temporally incoherent. Our method also outperforms

	Skeletal motion				Detail preserv.		
Context duration	MPJPE (m)↓	PA-MPJPE (m)↓	Acc↓	PA-Acc↓	MPVD (m)↓	PA-MPVD (m)↓	MEDL (mm)↓
0.16s (5 frames)	0.203	0.073	0.021	0.009	0.158	0.061	0.505
0.33s (10 frames)	0.167	0.050	0.021	0.008	0.137	0.045	0.507
0.5s(15 frames)	0.161	0.046	0.020	0.008	0.136	0.044	0.519
1s (30 frames)	0.158	0.043	0.021	0.008	0.134	0.040	0.524
2s (60 frames)	0.159	0.042	0.023	0.008	0.136	0.041	0.522

Table 6.3: Learning with different temporal contexts on SMPL test set. Training with long-term context of 1s improves the results.

the state-of-the-art motion prior H4D on almost all evaluation metrics on both datasets. In particular, the skeletal joint positions after Procrustes alignment are significantly more accurate for both test sets, and without Procrustes alignment, the mean is $4.9cm$ more accurate for naked target shapes, while being almost identical ($2mm$ worse) for clothed ones. Joint accelerations are more accurate when using our model. Geometric detail is significantly better preserved using our model when considering Procrustes alignment for both datasets, and without Procrustes alignment, the errors of both models are similar. This implies that our model retains geometric detail better, but that global alignment is not perfect.

Comparison to a retargeting method using correspondences As we outperform correspondence free retargeting methods by a large margin, we also compare our method to TST [89], a state-of-the-art deformation transfer method considering short-term temporal dynamics that requires point-to-point correspondences for both training and inference and provide this method with correspondences. Our method only performs slightly worse than TST on the evaluation metrics; Table 6.4 shows that all errors are within $2cm$ of TST for both test sets. This performance, which is close to a state-of-the-art method leveraging correspondences, highlights the potential of our correspondence-free method.

	Skeletal motion				Detail preserv.		
	MPJPE (m) ↓	PA-MPJPE (m) ↓	Acc ↓	PA-Acc ↓	MPVD (m) ↓	PA-MPVD (m) ↓	MDEL (mm) ↓
Naked target shapes from SMPL test set							
TST [89]	0.152	0.028	0.005	0.004	0.130	0.028	0.866
Ours	0.158	0.043	0.021	0.008	0.134	0.040	0.524
Clothed target shapes from CAPE test set							
TST [89]	0.096	0.027	0.005	0.004	0.064	0.028	0.953
Ours	0.105	0.040	0.023	0.008	0.075	0.038	0.539

Table 6.4: Comparison of our correspondence-free method to a state-of-the-art method which need correspondences. The comparison is done on naked (top) and clothed (bottom) target shapes.

6.3.3 Limitations

While our method gives state-of-the-art results for the correspondence-free motion retargeting problem, limitations remain. We cannot generalize on clothed shapes with wide garments which is due to LBS limitations. The method is also restricted to shapes that can be parameterized by skeleton with fixed topology and is not able to capture detailed hand or facial motion.

6.3.4 Ablations

We now provide ablations for the three different steps of the method. For the skeleton regressor and the skinning prior, ablations are conducted on the AMASS test set which provides ground truths. For the unsupervised skeletal retargeting, no skeletal ground truth is available so we evaluate on retargeting densely sampled geometry on the SMPL test set.

Skeleton regressor For the first module, we compare between a PointNet [136] architecture that considers global features and the PointFormer [128] architecture that introduces local features. Using PointFormer improves the MPJPE on the AMASS test set to $21mm$ over $46mm$ for PointNet. Figure 6.3 shows the regression result on a challenging pose from the AMASS test set. The PointNet based model which only leverages global features generalizes poorly while the PointFormer based model outputs more precise joint positions.

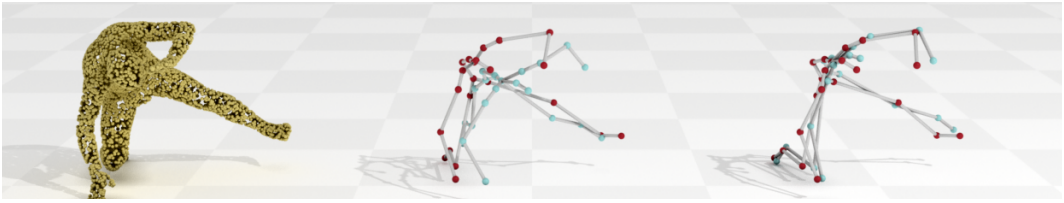


Figure 6.3: Comparison of the joint regression on a challenging pose from the AMASS test set (left). The ground truth joint positions are shown in red, the predicted joint positions in blue. The regression of the PointNet based model (middle) is imprecise for all joints. The PointFormer based model regression (right) only has a noticeable error on the right wrist and right ankle joints

Skeletal retargeting For the second module, we show that using the 6D rotation representation and the rotation supervision using $\mathcal{L}_{rot}$ improve the overall retargeting. Figure 6.4 shows the different retargeting results on a challenging HipHop motion from a female shape to a target male shape. The transition from quaternion to 6D rotation leads to a significant improvement on the retargeting on almost all metrics as the quaternion representation is more prone to generate unrealistic twists. Supervising with $\mathcal{L}_{rot}$ also leads to an improvement in the geometric detail preservation around the feet, wrists and head joints. This is because these joints are leaf joints of the hierarchical skeleton, so they have no influence on the joint-based loss $\mathcal{L}_{cycon}$, but they do influence $\mathcal{L}_{rot}$.

	Skeletal motion				Detail preserv.		
	MPJPE (m)	PA-MPJPE (m)	Acc	PA-Acc	MPVD (m)	PA-MPVD (m)	MEDL (m)
Quaternion ([77])	0.163	0.056	0.023	0.011	0.165	0.089	1.597
Quaternion + $\mathcal{L}_{rot}$	0.161	0.044	0.021	0.008	0.156	0.076	1.445
6D	0.159	0.043	0.024	0.008	0.137	0.048	0.684
6D + $\mathcal{L}_{rot}$	0.158	0.043	0.021	0.008	0.134	0.040	0.524

Table 6.5: Quantitative evaluation on SMPL shapes with different rotation representation for θ and ablation of the rotation cycle consistency loss $\mathcal{L}_{rot}$. Using $\mathcal{L}_{rot}$ and the 6D representation lead to an improvement.

Skinning prior For the third module, using pose-corrective offsets improves skinning precision by reducing the average reconstruction error on AMASS test set from $8.4mm$ to $5mm$.

6.3.5 Animating target shape with captured 4D data

We finally demonstrate our method's performance when animating a target shape with the raw 4D output of a multi-view acquisition platform. To this end, we

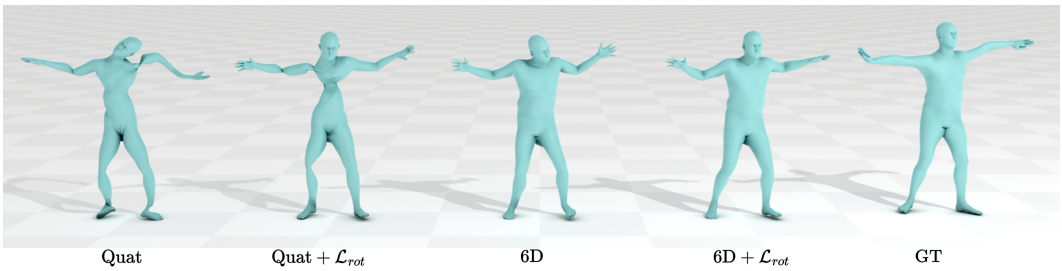


Figure 6.4: Visualization of a challenging pose from the retargeting result of an HipHop motion from a female shape to a male shape. Quaternion representation is prone to predict unrealistic twist, introducing $\mathcal{L}_{rot}$ improves the head and feet retargeting

take sequences of the multi-view test set and directly retarget them to characters generated using SMPL, CAPE, and a Mixamo character designed using computer aided design (CAD) tools, respectively. Note that the input sequence suffers from acquisition noise and that no correspondence information is available, *i.e.*we input the raw untracked 4D sequence.

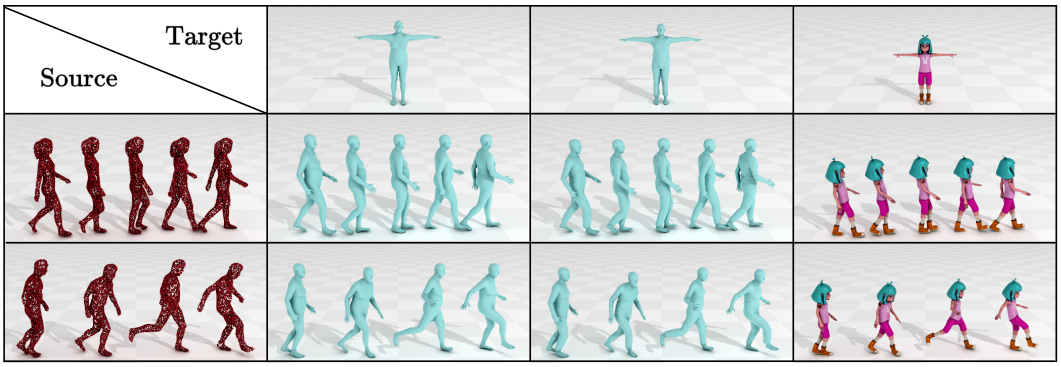


Figure 6.5: Animating target shapes with untracked captured 4D data directly. We consider a walking motion (top) and a kicking motion (bottom), which are retargeted to a naked (left), clothed (middle) and a CAD-generated (right) target shape.

Fig. 6.5 shows the results obtained using our method. Note how the motion of the source sequence as well as the geometric detail of the different target shapes are preserved by our method. To the best of our knowledge, our method is the first that can retarget untracked 4D acquisition data online.

These examples show the robustness of our method to unseen shapes. The first source motion exhibits a body shape with hair, not seen during training, demonstrating the robustness of our method to unseen source shapes. The preservation of the geometry for CAPE and CAD generated target shapes also demonstrates that our model generalizes well when considering unseen target shapes.

Quantitative and qualitative results show our method to generalize well on unseen motions *e.g.*from Mixamo and unseen shapes *e.g.*clothed shapes from CAPE and untracked 4D output of multi-view acquisition platforms.

6.4 Conclusion

In this chapter, we proposed the first online retargeting method that allows to animate a target shape with a correspondence-free source motion. We demonstrated that including long term temporal context of $1s$ is beneficial when retargeting dense motion. Our low dimensional intermediate skeletal representation combined with the skinning field generalizes well to unseen shapes and motions. In particular, we demonstrate that our model, learned exclusively on naked body shapes, generalizes to inputs with hair and clothing.

Interesting future works include going beyond linear blend skinning to allow for extensions to complex garments such as wide or layered clothing. One option is to explicitly include clothing in the model.

7 Summary and extensions

7.1 Résumé et futurs axes de recherche

Cette thèse a présenté plusieurs contributions à la représentation et à la synthèse des mouvements humains en 3D. Dans le chapitre 4, nous avons d'abord présenté une représentation générative du mouvement basée sur un espace latent structuré qui démêle les informations sur le mouvement des informations sur la morphologie. En utilisant cette représentation, nous avons montré qu'il était possible de générer de nouveaux mouvements par interpolation dans l'espace latent, de transférer de manière réaliste des mouvements vers de nouvelles morphologies et d'aligner spatialement et de densifier des observations de nuages de points éparses acquises à l'aide d'une plate-forme multi-vues.

Au chapitre 4, le mouvement humain est caractérisé comme une séquence discrète de corps utilisant un nombre fixe d'images d'ancrage et aligné dans l'espace à l'aide d'une déformation temporelle dynamique limitant la variété des mouvements considérés. Au chapitre 5, nous avons étendu cette représentation pour traiter des mouvements plus longs avec plus de variabilité en adoptant une représentation latente séquentielle qui tient compte de la complexité du mouvement dans une séquence de primitives latentes. Nous avons également amélioré l'architecture du réseau pour fournir une représentation continue dans le temps à l'aide d'une fonction de décodage implicite et d'un transformer de séquence à séquence plus flexible pour traiter les longueurs d'entrée variables. Cette représentation séquentielle a montré des performances de pointe sur une tâche d'achèvement de mouvement sans correspondance, se généralisant à des mouvements avec des poses complexes comme une roue de charrette.

Enfin, dans le chapitre 6, nous avons exploré la tâche spécifique de reciblage de mouvement pour synthétiser de nouveaux mouvements d'un personnage cible en utilisant le mouvement d'un personnage source. Nous avons proposé une nouvelle approche qui fonctionne en ligne et sans correspondance. Notre méthode diffère de l'état de l'art en exploitant efficacement les informations temporelles du mouvement source au niveau du squelette tout en préservant les détails géométriques à l'aide

d'un champ de skinning dense.

Au travers de ces trois chapitres, nous avons donc proposé différentes manières de générer et de représenter le mouvement humain en 3D, et nous avons montré que l'exploitation du contexte temporel est bénéfique pour désambiguïser les données d'entrée, en particulier dans les contextes sans correspondance, que ce soit dans la tâche d'achèvement ou dans la tâche de reciblage. Nous avons également montré que les modèles entraînés sur des données synthétiques sont robustes au bruit et peuvent être appliqués à des acquisitions réelles réalisées à l'aide d'une plateforme multi-vues.

Tout au long de cette thèse, nous avons supposé que la morphologie restait constante tout au long du mouvement. En étendant l'espace de forme statique des modèles corporels pour gérer la morphologie variable en ajoutant une dimension temporelle à la représentation de la morphologie, le réalisme des mouvements générés pourrait être amélioré pour mieux gérer les déformations des tissus mous et des vêtements. Pour résoudre le problème des données, il est possible d'exploiter la grande quantité de données de mouvement RGB 2D accessibles qui peuvent être converties en mouvement 3D à l'aide de champs de radiance neuronaux [17] ou de modèles génératifs d'image en 3D tels que [137, 138].

L'architecture des autoencodeurs variationnels proposée dans les chapitres 4 et 5 est limitée pour créer un modèle génératif pour les mouvements longs sans restriction sur la variété des poses. Bien que le chapitre 5 montre que le bon pouvoir représentatif sur les mouvements longs et complexes, il ne peut pas être facilement exploité pour générer de nouveaux mouvements par échantillonnage latent. D'après nos premières expériences dans ce sens, cela semble principalement dû à l'effondrement de la moyenne des VAE qui rend difficile la génération de mouvements qui diffèrent du mouvement moyen. Les récents modèles génératifs de textes et d'images basés sur la diffusion (par exemple [139]) sont plus prometteurs pour apprendre les représentations latentes des mouvements qui conservent les détails à haute fréquence et semblent moins sujettes à l'effondrement de la moyenne.

L'approche de reciblage proposée au chapitre 6 est une voie prometteuse pour le reciblage de mouvements sans correspondance. De nombreuses améliorations techniques sont possibles pour les trois blocs de cette méthode. À un niveau plus élevé, permettre à la forme cible de se déformer au fil du temps pourrait également améliorer considérablement le réalisme des formes habillées ou des formes comportant beaucoup de tissus mous. Une direction possible est de changer la configuration du problème et de considérer une séquence de nuages de points de la cible au lieu d'une T-Pose de la cible, ce qui permettrait d'extraire des caractéristiques dynamiques de la cible.

7.2 Summary

This thesis presented several contributions to 3D human motion representation and synthesis. In Chapter 4, we first introduced a task generic generative representation of motion based on a structured latent space that disentangles the motion information from the morphology information. Using this representation, we showed that it was possible to generate new motion by interpolating in the latent

space, to realistically transfer motions to new morphologies and to spatially align and densify sparse point cloud observations acquired using a multi-view platform.

In Chapter 4, human motion is characterized as a discrete sequence of bodies using a fixed number of anchor frames and spatially aligned using dynamic time warping limiting the variety of the considered motions. In Chapter 5, we extended this representation to handle longer motion with more variability by adopting a sequential latent representation which factors the complexity of the motion in a sequence of latent primitives. We also improved the network architecture to provide a representation continuous in time using an implicit decoder function and a more flexible sequence to sequence transformer to handle variable input lengths. This sequential representation showed state-of-the-art performances on a correspondence-less motion completion task, generalizing to motion with complex poses such as a cartwheel.

Finally, in Chapter 6, we explored the specific task of motion retargeting to synthesize new motions of a target character using the motion of a source character. We proposed a new approach that operates online and without correspondence. Our method differs from the state-of the art by efficiently leveraging the temporal information of the source motion at a skeletal level while preserving the geometric details using a dense skinning field.

Through these three chapters, we therefore proposed different ways of generating and representing 3D human motion, and showed that leveraging temporal context is beneficial to disambiguate the input data, especially in correspondence-less settings either in the completion task or in the retargeting task. We also showed that data-driven models trained on synthetic data are robust to noise and can be applied to real acquisitions captured using a multi-view platform.

7.3 Extensions

7.3.1 Geometric details and clothing

Throughout this thesis, we assumed morphology to remain constant through the motion. By extending the static shape space of body models to handle varying morphology by adding a temporal dimension to the morphology representation, the realism of the generated motions could be improved to better handle soft-tissue deformations and clothing. A major limitation to explore this direction at this time is the absence of a large enough dataset of clothed human in motion. A possibility to address the data problem is to leverage the large amount of accessible 2D RGB motion data that may be converted to 3D motion using neural radiance fields [17] or image to 3D generative models such as [137, 138].

7.3.2 Latent motion representations

The variational autoencoders architecture proposed in Chapters 4 and 5 is limiting to create a generative model for long motion with no restriction on the pose variety. While Chapter 5 shows that the good representative power on long and complex motion, it cannot be easily leveraged to generate new motions by latent sampling. From our first experiments in this direction, it seems mostly due to the mean

collapse of the VAEs which makes it hard to generate motions that differ from the mean motion. The recent text to image generative models based on diffusion (e.g. [139]) are more promising to learn latent representations of motions that retain high-frequency details and seem less prone to mean collapse.

7.3.3 Motion retargeting

The proposed retargeting approach introduced in Chapter 6 is a promising direction for correspondence-free motion retargeting. There are many technical improvements possible for the three blocks of this method. At a higher level, allowing the target shape to deform through time could also greatly improve the realism on clothed shapes or shapes with a lot of soft-tissues. A possible direction is to change the problem setting and consider a target point cloud sequence instead of a target T-Pose, allowing to extract dynamic features of the target.

Funding & Supervision

This work was supported by French government funding managed by the National Research Agency under the Investments for the Future program (PIA) grant ANR-21-ESRE-0030 (CONTINUUM) and 3DMOVE - 19- CE23-0013-01.

This research was conducted inside the MORPHEO team of the INRIA Rhones Al--pes which focuses on the capture and analysis of 3D shapes in motion.

Potential negative societal impact

This work presented method that allows for automated generation of motion and long-term and geometrically detailed motion retargeting between different digitized human models. It could be used without the consent of the user to animate static 3D scans, or even 3D reconstructions generated from 2D images, *e.g.* to generate disinformation.

Bibliography

[1] V. Leroy, J.-S. Franco, and E. Boyer, "Shape reconstruction using volume sweeping and learned photoconsistency," in European Conference on Computer Vision, pp. 781–796, 2018.

[2] J. W. Tukey, "Box-and-whisker plots," Exploratory data analysis, pp. 39–43, 1977.

[3] D. Terzopoulos and K. Waters, "Physically-based facial modelling, analysis, and animation," The journal of visualization and computer animation, vol. 1, no. 2, pp. 73–80, 1990.

[4] M. Perse, J. Pers, M. Kristan, S. Kovacic, and G. Vuckovic, "Physics-based modelling of human motion using kalman filter and collision avoidance algorithm," in ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005., pp. 328–333, IEEE, 2005.

[5] M. A. R. Ahad, J. Tan, H. Kim, and S. Ishikawa, "Action dataset—a survey," in SICE Annual Conference 2011, pp. 1650–1655, IEEE, 2011.

[6] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "Amass: Archive of motion capture as surface shapes," in International Conference on Computer Vision, pp. 5442–5451, 2019.

[7] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in European conference on computer vision, pp. 501–518, Springer, 2016.

[8] F. Wang, S. Galliani, C. Vogel, P. Speciale, and M. Pollefeys, "Patchmatchnet: Learned multi-view patchmatch stereo," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14194–14203, 2021.

[9] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "Scape: shape completion and animation of people," in ACM SIGGRAPH 2005 Papers, pp. 408–416, 2005.

[10] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: a skinned multi-person linear model," Transactions on Graphics, vol. 34, no. 6, pp. 1–16, 2015.

[11] N. F. Troje, "Retrieving information from human movement patterns," Understanding events: How humans see, represent, and act on events, vol. 1, pp. 308–334, 2008.

[12] N. F. Troje, "Decomposing biological motion: A framework for analysis and synthesis of human gait patterns," Journal of vision, vol. 2, no. 5, pp. 2–2, 2002.

[13] M. Marsot, S. Wuhrer, J.-S. Franco, and S. Durocher, "A structured latent space for human body motion generation," in Conference on 3D Vision, 2022.

[14] M. Marsot, S. Wuhrer, J.-S. Franco, and A. H. Olivier, "Representing motion as a sequence of latent primitives, a flexible approach for human motion modelling." working paper or preprint, Sept. 2022.

[15] M. Marsot, R. Rekik, S. Wuhrer, J.-S. Franco, and A.-H. Olivier, "Correspondence-free online human motion retargeting," 2023.

[16] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, "Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization," in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2304–2314, 2019.

[17] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," Communications of the ACM, vol. 65, no. 1, pp. 99–106, 2021.

[18] M. Keller, S. Zuffi, M. J. Black, and S. Pujades, "Osso: Obtaining skeletal shape from outside," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 20492–20501, 2022.

[19] Y. Lipman, O. Sorkine, D. Levin, and D. Cohen-Or, "Linear rotation-invariant coordinates for meshes," Transactions on Graphics, vol. 24, no. 3, pp. 479–487, 2005.

[20] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, and H.-Y. Shum, "Mesh editing with poisson-based gradient field manipulation," in SIGGRAPH, pp. 644–651, 2004.

[21] L. Kavan, S. Collins, J. Žára, and C. O'Sullivan, "Skinning with dual quaternions," in Proceedings of the 2007 symposium on Interactive 3D graphics and games, pp. 39–46, 2007.

[22] P. Joshi, W. C. Tien, M. Desbrun, and F. Pighin, "Learning controls for blend shape based realistic facial animation," in Symposium of Computer Animation, pp. 17–20, 2003.

[23] P. G. Kry, D. L. James, and D. K. Pai, "Eigenskin: real time large deformation character skinning in hardware," in Symposium on Computer Animation, pp. 153–159, 2002.

[24] I. Baran and J. Popović, "Automatic rigging and animation of 3d characters," ACM Transactions on graphics (TOG), vol. 26, no. 3, pp. 72–es, 2007.

[25] R. Wareham and J. Lasenby, "Bone glow: An improved method for the assignment of weights for mesh deformation," in International Conference on Articulated Motion and Deformable Objects, pp. 63–71, Springer, 2008.

[26] Z. Xu, Y. Zhou, E. Kalogerakis, C. Landreth, and K. Singh, "Rignet: Neural rigging for articulated characters," Transactions on Graphics, vol. 39, no. 4, pp. 58:1–14, 2020.

[27] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," arXiv preprint arXiv:1312.6203, 2013.

[28] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," Advances in neural information processing systems, vol. 29, 2016.

[29] N. Verma, E. Boyer, and J. Verbeek, "Feastnet: Feature-steered graph convolutions for 3d shape analysis," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2598–2606, 2018.

[30] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," Acm Transactions On Graphics (tog), vol. 38, no. 5, pp. 1–12, 2019.

[31] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 4460–4470, 2019.

[32] P. Zins, Y. Xu, E. Boyer, S. Wuhrer, and T. Tung, "Data-driven 3d reconstruction of dressed humans from sparse views," in 2021 International Conference on 3D Vision (3DV), pp. 494–504, IEEE, 2021.

[33] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," ACM siggraph computer graphics, vol. 21, no. 4, pp. 163–169, 1987.

[34] B. Deng, J. P. Lewis, T. Jeruzalski, G. Pons-Moll, G. Hinton, M. Norouzi, and A. Tagliasacchi, "Nasa neural articulated shape approximation," in European Conference on Computer Vision, pp. 612–628, Springer, 2020.

[35] R. W. Sumner and J. Popović, "Deformation transfer for triangle meshes," Transactions on Graphics, vol. 23, no. 3, pp. 399–405, 2004.

[36] I. Baran, D. Vlasic, E. Grinspun, and J. Popović, "Semantic deformation transfer," in SIGGRAPH, pp. 1–6, 2009.

[37] K. Zhou, W. Xu, Y. Tong, and M. Desbrun, "Deformation transfer to multi-component objects," in Computer Graphics Forum, vol. 29, pp. 319–325, 2010.

[38] A. Boukhayma, J.-S. Franco, and E. Boyer, "Surface motion capture transfer with gaussian process regression," in Conference on Computer Vision and Pattern Recognition, pp. 184–192, 2017.

[39] J. Wang, C. Wen, Y. Fu, H. Lin, T. Zou, X. Xue, and Y. Zhang, "Neural pose transfer by spatially adaptive instance normalization," in Conference on Computer Vision and Pattern Recognition, pp. 5831–5839, 2020.

[40] J. Basset, A. Boukhayma, S. Wuhrer, F. Multon, and E. Boyer, "Neural human deformation transfer," in International Conference on 3D Vision, pp. 545–554, 2021.

[41] K. Zhou, B. L. Bhatnagar, and G. Pons-Moll, "Unsupervised shape and pose disentanglement for 3d meshes," in European Conference on Computer Vision, pp. 341–357, 2020.

[42] L. Cosmo, A. Norelli, O. Halimi, R. Kimmel, and E. Rodola, "Limp: Learning latent shape representations with metric preservation priors," in European Conference on Computer Vision, pp. 19–35, Springer, 2020.

[43] B. Allen, B. Curless, and Z. Popović, "The space of human body shapes: reconstruction and parameterization from range scans," Transactions on Graphics, vol. 22, no. 3, pp. 587–594, 2003.

[44] A. Neophytou and A. Hilton, "Shape and pose space deformation for subject specific animation," in Conference on 3D Vision, pp. 334–341, IEEE, 2013.

[45] L. Pishchulin, S. Wuhrer, T. Helten, C. Theobalt, and B. Schiele, "Building statistical shape spaces for 3d human modeling," Pattern Recognit., vol. 67, pp. 276–286, 2017.

[46] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black, "Expressive body capture: 3d hands, face, and body from a single image," in Conference on Computer Vision and Pattern Recognition, 2019.

[47] H. Xu, E. G. Bazavan, A. Zanfir, W. T. Freeman, R. Sukthankar, and C. Sminchisescu, "Ghum & ghuml: Generative 3d human shape and articulated pose models," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6184–6193, 2020.

[48] G. Pons-Moll, J. Romero, N. Mahmood, and M. J. Black, "Dyna: A model of dynamic human shape in motion," ACM Transactions on Graphics (TOG), vol. 34, no. 4, pp. 1–14, 2015.

[49] I. Santesteban, E. Garces, M. A. Otaduy, and D. Casas, "Softsmpl: Data-driven modeling of nonlinear soft-tissue dynamics for parametric humans," in Computer Graphics Forum, vol. 39, pp. 65–75, Wiley Online Library, 2020.

[50] L. Cosmo, A. Norelli, O. Halimi, R. Kimmel, and E. Rodola, "Limp: Learning latent shape representations with metric preservation priors," in European Conference on Computer Vision, pp. 19–35, Springer, 2020.

[51] B. Jiang, J. Zhang, J. Cai, and J. Zheng, "Disentangled human body embedding based on deep hierarchical neural network," IEEE transactions on visualization and computer graphics, vol. 26, no. 8, pp. 2560–2575, 2020.

[52] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," arXiv preprint arXiv:1312.6114, 2013.

[53] A. Davydov, A. Remizova, V. Constantin, S. Honari, M. Salzmann, and P. Fua, "Adversarial parametric pose prior," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10997–11005, 2022.

[54] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," Advances in neural information processing systems, vol. 27, 2014.

[55] G. Tiwari, D. Antić, J. E. Lenssen, N. Sarafianos, T. Tung, and G. Pons-Moll, "Pose-ndf: Modeling human pose manifolds with neural distance fields," in European Conference on Computer Vision, pp. 572–589, Springer, 2022.

[56] T. Alldieck, H. Xu, and C. Sminchisescu, "imghum: Implicit generative models of 3d human shape and articulated pose," in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5461–5470, 2021.

[57] M. Mihajlovic, Y. Zhang, M. J. Black, and S. Tang, "Leap: Learning articulated occupancy of people," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10461–10471, 2021.

[58] C. Rose, M. F. Cohen, and B. Bodenheimer, "Verbs and adverbs: Multidimensional motion interpolation," IEEE Computer Graphics and Applications, vol. 18, no. 5, pp. 32–40, 1998.

[59] N. F. Troje, "Decomposing biological motion: A framework for analysis and synthesis of human gait patterns," Journal of vision, vol. 2, no. 5, pp. 2–2, 2002.

[60] D. Holden, J. Saito, and T. Komura, "A deep learning framework for character motion synthesis and editing," ACM Transactions on Graphics (TOG), vol. 35, no. 4, pp. 1–11, 2016.

[61] S. Lohit, R. Anirudh, and P. Turaga, "Recovering trajectories of unmarked joints in 3d human actions using latent space optimization," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2342–2351, 2021.

[62] M. Petrovich, M. J. Black, and G. Varol, "Action-conditioned 3d human motion synthesis with transformer VAE," in International Conference on Computer Vision, 2021.

[63] N. Athanasiou, M. Petrovich, M. J. Black, and G. Varol, "Teach: Temporal action composition for 3d humans," arXiv preprint arXiv:2209.04066, 2022.

[64] D. Rempe, T. Birdal, A. Hertzmann, J. Yang, S. Sridhar, and L. J. Guibas, "Humor: 3d human motion model for robust pose estimation," in International Conference on Computer Vision, pp. 11488–11499, 2021.

[65] Y. Lee, K. Wampler, G. Bernstein, J. Popović, and Z. Popović, "Motion fields for interactive character locomotion," in ACM SIGGRAPH Asia 2010 papers, pp. 1–8, 2010.

[66] J. Li, R. Villegas, D. Ceylan, J. Yang, Z. Kuang, H. Li, and Y. Zhao, "Task-generic hierarchical human motion prior using vaes," in 2021 International Conference on 3D Vision (3DV), pp. 771–781, IEEE, 2021.

[67] J. Xu, M. Wang, J. Gong, W. Liu, C. Qian, Y. Xie, and L. Ma, "Exploring versatile prior for human motion via motion frequency guidance," in 2021 International Conference on 3D Vision (3DV), pp. 606–616, IEEE, 2021.

[68] S. Ghorbani, C. Wloka, A. Etemad, M. A. Brubaker, and N. F. Troje, "Probabilistic character motion synthesis using a hierarchical deep latent variable model," in Computer Graphics Forum, vol. 39, pp. 225–239, 2020.

[69] C. He, J. Saito, J. Zachary, H. Rushmeier, and Y. Zhou, "Nemf: Neural motion fields for kinematic animation," arXiv preprint arXiv:2206.03287, 2022.

[70] M. Gleicher, "Retargetting motion to new characters," in SIGGRAPH, pp. 33–42, 1998.

[71] J. Lee and S. Y. Shin, "A hierarchical approach to interactive motion editing for human-like figures," in SIGGRAPH, pp. 39–48, 1999.

[72] K.-J. Choi and H.-S. Ko, "Online motion retargetting," The Journal of Visualization and Computer Animation, vol. 11, no. 5, pp. 223–235, 2000.

[73] S. Tak and H.-S. Ko, "A physically-based motion retargeting filter," Transactions on Graphics, vol. 24, no. 1, pp. 98–117, 2005.

[74] H.-S. Ko, K.-J. Choi, M. G. Choi, S. Tak, B. Choe, and O.-Y. Song, "Research problems for creating digital actors.," in Eurographics State of the Art Reports, 2003.

[75] B. Delhaisse, D. Esteban, L. Rozo, and D. Caldwell, "Transfer learning of shared latent spaces between robots with similar kinematic structure," in International Joint Conference on Neural Networks, pp. 4142–4149, 2017.

[76] H. Jang, B. Kwon, M. Yu, S. U. Kim, and J. Kim, "A variational u-net for motion retargeting," in SIGGRAPH Asia Posters, pp. 1–2, 2018.

[77] R. Villegas, J. Yang, D. Ceylan, and H. Lee, "Neural kinematic networks for unsupervised motion retargetting," in Conference on Computer Vision and Pattern Recognition, pp. 8639–8648, 2018.

[78] K. Aberman, P. Li, D. Lischinski, O. Sorkine-Hornung, D. Cohen-Or, and B. Chen, "Skeleton-aware networks for deep motion retargeting," Transactions on Graphics, vol. 39, no. 4, pp. 62–1, 2020.

[79] A. Kuznetsova, N. Troje, and B. Rosenhahn, "A statistical model for coupled human shape and motion synthesis," in Conf. Comput. Graph. Theory App., pp. 227–236, 2013.

[80] I. Akhter, T. Simon, S. Khan, I. Matthews, and Y. Skeikh, "Bilinear spatiotemporal basis models," ToG, vol. 31, no. 2, pp. #17:1–12, 2012.

[81] A. Boukhayma and E. Boyer, "Surface motion capture animation synthesis," TVCG, vol. 25, no. 6, pp. 2270–2283, 2018.

[82] J. Regateiro, A. Hilton, and M. Volino, "Dynamic surface animation using generative networks," in 3DV, pp. 376–385, IEEE, 2019.

[83] J. Regateiro, M. Volino, and A. Hilton, "Deep4d: A compact generative representation for volumetric video," Front. Virtual Reality, vol. 2, p. 739010, 2021.

[84] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Occupancy flow: 4d reconstruction by learning particle dynamics," in Proceedings of the IEEE/CVF international conference on computer vision, pp. 5379–5389, 2019.

[85] B. Jiang, Y. Zhang, X. Wei, X. Xue, and Y. Fu, "Learning compositional representation for 4d captures with neural ODE," in Conference on Computer Vision and Pattern Recognition, pp. 5340–5350, 2021.

[86] B. Jiang, Y. Zhang, X. Wei, X. Xue, and Y. Fu, "H4D: human 4d modeling by learning neural compositional representation," in Conference on Computer Vision and Pattern Recognition, pp. 19355–19365, 2022.

[87] R. Villegas, D. Ceylan, A. Hertzmann, J. Yang, and J. Saito, "Contact-aware retargeting of skinned motion," in International Conference on Computer Vision, 2021.

[88] H. Chen, H. Tang, N. Sebe, and G. Zhao, "Aniformer: Data-driven 3d animation with transformer," in British Machine Vision Conference, 2021.

[89] J. Regateiro and E. Boyer, "Temporal shape transfer network for 3d human motion," in International Conference on 3D Vision, 2022.

[90] K. M. Robinette, S. Blackwell, H. Daanen, M. Boehmer, and S. Fleming, "Civilian american and european surface anthropometry resource (caesar), final report. volume 1. summary," tech. rep., Sytronics Inc Dayton Oh, 2002.

[91] F. Bogo, J. Romero, M. Loper, and M. J. Black, "Faust: Dataset and evaluation for 3d mesh registration," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3794–3801, 2014.

[92] T. Magnenat, R. Laperrière, and D. Thalmann, "Joint-dependent local deformations for hand animation and object grasping," tech. rep., Canadian Inf. Process. Soc, 1988.

[93] F. Cordier and N. Magnenat-Thalmann, "A data-driven approach for real-time clothes simulation," in 12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings., pp. 257–266, IEEE, 2004.

[94] L. Euler, "Nova methodus motum corporum rigidorum degerminandi," Novi commentarii academiae scientiarum Petropolitanae, pp. 208–238, 1776.

[95] D. H. Ballard, "Modular learning in neural networks.," in Aaai, vol. 647, pp. 279–284, 1987.

[96] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," Journal of the royal statistical society: series B (methodological), vol. 39, no. 1, pp. 1–22, 1977.

[97] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," 2016.

[98] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," nature, vol. 323, no. 6088, pp. 533–536, 1986.

[99] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, et al., "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[100] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[101] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," Neural computation, vol. 12, no. 10, pp. 2451–2471, 2000.

[102] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014.

[103] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017.

[104] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," arXiv preprint arXiv:1409.0473, 2014.

[105] Y. Kim, C. Denton, L. Hoang, and A. M. Rush, "Structured attention networks," arXiv preprint arXiv:1702.00887, 2017.

[106] Y. Chen, Z. Yang, X. Zheng, Y. Chang, and X. Li, "Pointformer: A dual perception attention-based network for point cloud classification," in Proceedings of the Asian Conference on Computer Vision, pp. 3291–3307, 2022.

[107] J. Won and J. Lee, "Learning body shape variation in physics-based characters," ACM Transactions on Graphics (TOG), vol. 38, no. 6, pp. 1–12, 2019.

[108] L. Sigal, D. J. Fleet, N. F. Troje, and M. Livne, "Human attributes from 3d pose tracking," in European conference on computer vision, pp. 243–257, Springer, 2010.

[109] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2891–2900, 2017.

[110] B. Zhou, J.-S. Franco, F. Bogo, B. Tekin, and E. Boyer, "Reconstructing human body mesh from point clouds by adversarial gp network," in Proceedings of the Asian Conference on Computer Vision, 2020.

[111] J. Yang, J.-S. Franco, F. Hétroy-Wheeler, and S. Wuhrer, "Estimation of human body shape in motion with wide clothing," in European Conference on Computer Vision, pp. 439–454, Springer, 2016.

[112] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5745–5753, 2019.

[113] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," Advances in neural information processing systems, vol. 28, 2015.

[114] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in International conference on machine learning, pp. 794–803, PMLR, 2018.

[115] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series.," in KDD workshop, vol. 10, pp. 359–370, Seattle, WA, USA:, 1994.

[116] K. Shoemake, "Animating rotation with quaternion curves," in Proceedings of the 12th annual conference on Computer graphics and interactive techniques, pp. 245–254, 1985.

[117] J. Li, R. Villegas, D. Ceylan, J. Yang, Z. Kuang, H. Li, and Y. Zhao, "Task-generic hierarchical human motion prior using vaes," in 2021 International Conference on 3D Vision (3DV), pp. 771–781, IEEE, 2021.

[118] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in Conference on Computer Vision and Pattern Recognition, 2019.

[119] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in International Conference on Learning Representations, 2015.

[120] F. Bogo, J. Romero, G. Pons-Moll, and M. J. Black, "Dynamic faust: Registering human bodies in motion," in Conference on Computer Vision and Pattern Recognition, pp. 6233–6242, 2017.

[121] T. Von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3d human pose in the wild using imus and a moving camera," in European Conference on Computer Vision, pp. 601–617, 2018.

[122] J. Lim, H. J. Chang, and J. Y. Choi, "Pmnet: Learning of disentangled pose and movement for unsupervised motion retargeting.," in British Machine Vision Conference, no. 6:1–7, 2019.

[123] L. Liu, Y. Zheng, D. Tang, Y. Yuan, C. Fan, and K. Zhou, "Neuroskinning: Automatic skin binding for production characters with deep graph networks," Transactions on Graphics, vol. 38, no. 4, pp. 1–12, 2019.

[124] Z. Yang, S. Wang, S. Manivasagam, Z. Huang, W.-C. Ma, X. Yan, E. Yumer, and R. Urtasun, "S3: Neural shape, skeleton, and skinning fields for 3d human modeling," in Conference on Computer Vision and Pattern Recognition, pp. 13284–13293, 2021.

[125] A. Mosella-Montoro and J. Ruiz-Hidalgo, "Skinningnet: Two-stream graph convolutional neural network for skinning prediction of synthetic characters," in Conference on Computer Vision and Pattern Recognition, pp. 18593–18602, 2022.

[126] K. Aberman, R. Wu, D. Lischinski, B. Chen, and D. Cohen-Or, "Learning character-agnostic motion for motion retargeting in 2d," Transactions on Graphics, vol. 38, no. 4, pp. 75:1–14, 2019.

[127] H. Chen, H. Tang, H. Shi, W. Peng, N. Sebe, and G. Zhao, "Intrinsic-extrinsic preserved gans for unsupervised 3d pose transfer," in International Conference on Computer Vision, pp. 8630–8639, 2021.

[128] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in Proceedings of the IEEE/CVF international conference on computer vision, pp. 16259–16268, 2021.

[129] X. Ouyang and C. Feng, "Autoskin: Skeleton-based human skinning with deep neural networks," in Journal of Physics: Conference Series, vol. 1550, p. 032163, IOP Publishing, 2020.

[130] X. Chen, Y. Zheng, M. J. Black, O. Hilliges, and A. Geiger, "Snarf: Differentiable forward skinning for animating non-rigid neural implicit shapes," in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 11594–11604, 2021.

[131] B. Allen, B. Curless, Z. Popović, and A. Hertzmann, "Learning a correlated model of identity and pose-dependent body shape variation for real-time synthesis," in Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp. 147–156, Citeseer, 2006.

[132] B. Jiang, J. Zhang, Y. Hong, J. Luo, L. Liu, and H. Bao, "Bcnet: Learning body and cloth shape from a single image," in European Conference on Computer Vision, pp. 18–35, Springer, 2020.

[133] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, M. Elgharib, P. Fua, H.-P. Seidel, H. Rhodin, G. Pons-Moll, and C. Theobalt, "Xnect: Real-time multi-person 3d motion capture with a single rgb camera," Transactions On Graphics, vol. 39, no. 4, pp. 82–1, 2020.

[134] J. Y. Zhang, P. Felsen, A. Kanazawa, and J. Malik, "Predicting 3d human dynamics from video," in International Conference on Computer Vision, pp. 7114–7123, 2019.

[135] Q. Ma, J. Yang, A. Ranjan, S. Pujades, G. Pons-Moll, S. Tang, and M. J. Black, "Learning to dress 3d people in generative clothing," in Conference on Computer Vision and Pattern Recognition, pp. 6469–6478, 2020.

[136] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: deep learning on point sets for 3d classification and segmentation," in Conference on Computer Vision and Pattern Recognition, 2017.

[137] Z. Zheng, T. Yu, Y. Liu, and Q. Dai, "Pamir: Parametric model-conditioned implicit representation for image-based human reconstruction," IEEE transactions on pattern analysis and machine intelligence, vol. 44, no. 6, pp. 3170–3184, 2021.

[138] S. Saito, T. Simon, J. Saragih, and H. Joo, "Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 84–93, 2020.

[139] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10684–10695, 2022.