Representing motion as a sequence of latent primitives, a flexible approach for human motion modelling

Mathieu Marsot, Stefanie Wuhrer, Jean-Sebastien Franco, Anne Hélène Olivier

Markerless performance capture using a sequence of latent primitives

Mathieu Marsot
Univ. Grenoble Alpes, Inria,
CNRS, Grenoble INP, LJK, 38000 Grenoble, France
mathieu.marsot@inria.fr

Stefanie Wuhrer
Univ. Grenoble Alpes, Inria,
CNRS, Grenoble INP, LJK, 38000 Grenoble, France
stefanie.wuhrer@inria.fr

Jean-Sebastien Franco
Univ. Grenoble Alpes, Inria,
CNRS, Grenoble INP, LJK, 38000 Grenoble, France
jean-sebastien.franco@inria.fr

Anne Hélène Olivier
Univ. Rennes, Inria, CNRS, IRISA, M2S, France
anne-helene.olivier@inria.fr

Abstract

We propose a markerless performance capture method that computes a temporally coherent 4D representation of an actor deforming over time from a sparsely sampled sequence of untracked 3D point clouds. Our method proceeds by latent optimization with a spatio-temporal motion prior. Recently, task generic motion priors have been introduced and propose a coherent representation of human motion based on a single latent code, with encouraging results with short sequences and given temporal correspondences. Extending these methods to longer sequences without correspondences is all but straightforward. One latent code proves inefficient to encode longer term variability, and latent space optimization will be very susceptible to erroneous local minima due to possible inverted pose fittings. We address both problems by learning a motion prior that encodes a 4D human motion sequence into a sequence of latent primitives instead of one latent code. We also propose an additional mapping encoder which directly projects a sequence of point clouds into the learned latent space to provide a good initialization of the latent representation at inference time. Our temporal decoding from latent space is implicit and continuous in time, providing flexibility with temporal resolution. We show experimentally that our method outperforms state-of-the-art motion priors.

1. Introduction

Markerless performance capture has practical applications in computer vision and graphics, in particular in the entertainment industry, *e.g.*, for 4D video editing or to render free-viewpoint video, and has been studied during the past decades. The goal is to compute a temporally coherent 4D representation of the 3D shape of a human actor deforming over time from untracked 2D or 3D input video. Existing studies consider data from different acquisition systems including multi-camera systems *e.g.* [29], depth cameras *e.g.* [3], and even monocular cameras *e.g.* [37]. Such capture output can typically be post-processed to obtain a 3D point cloud at each time frame. In this work, we consider as input a sequence of untracked 3D point clouds capturing a human body in motion of flexible duration, spatial and temporal sampling rate (*i.e.* input signals captured at arbitrary frame rate and spatial resolution), and output a temporally coherent 4D human motion sequence of arbitrarily dense frame rate that approximates the sparse input sequence.

Markerless performance capture is an inherently ill-posed problem that is challenging due to the large-scale and highly non-rigid deformations of the actor. A common solution is to rely on prior knowledge, either in the form of a template of the actor *e.g.* [37], or by using data-driven strategies that efficiently learn the space of human body shapes and poses, or even motions. Models that learn the configuration of static human bodies *e.g.* [23] have successfully been applied to extract temporally coherent human motion sequences when 3D point clouds *e.g.* [13] or 2D images *e.g.* [16] are available at every frame of the motion. More recently, data-driven motion priors for human bodies have been proposed and successfully applied to complete spatially or temporally incomplete inputs [36, 18]. These models learn from motions of fixed duration captured at a fixed frame rate and lead to impressive results when a good initialization or a sparse set of tracked markers over time are available, and with relatively short sequences.

However, it remains challenging to enhance sparsely sampled signals spatially and temporally, without initialization or temporally tracked markers, and process longer sequences.

In this work, we study the markerless performance capture problem with increasingly sparsely sampled signals in space and time, thereby addressing a fundamental problem related to the robustness to the acquisition process. In particular, we investigate how spatio-temporal motion priors can help with the processing of highly degraded input signals. We introduce a new data-driven spatio-temporal motion prior and demonstrate its suitability to complete sparsely sampled untracked input sequences. Our representation encodes a 4D human motion sequence into a sequence of latent primitives. Inspired by the flexibility of sequence-to-sequence (seq2seq) architectures w.r.t. the sequence length given as input [33], we learn this representation using a transformer approach. This allows to generalize to inputs of varying duration, different frame rates and different motions. This part of our model acts as an encoder, and maps a 4D motion sequence into a sequence of latent primitives. Sequencing the representation helps to generalize better for longer motion as long motions are often better represented as a succession of actions rather than as a single element. To allow for arbitrarily densely sampled outputs in time, we decode the sequence of latent primitives using a decoder that is implicit in time and outputs a parametric 3D human body model [19] for a given time instant.

The training of the encoder builds on a parametric 3D human body model to reduce the dimensionality of the input sequence. However, as parametric input models are not available during inference, where the method processes noisy point clouds, we additionally learn a mapping function from the space of sparsely sampled point cloud sequences to the space of latent primitive sequences. This allows to reconstruct a parametric 4D human body motion sequence from untracked inputs sparsely sampled in space and time during inference, by first mapping the point cloud sequence to a latent primitive sequence and by subsequently decoding parametric 3D human body models at an arbitrary frame rate. To summarize, our main contributions are

- A novel motion representation using a sequence of latent primitives.
- Two encoder training, allowing the method to simultaneously build a consistent latent space of parametric models and a projection to the latent space from sparse point clouds.
- An implicit representation of the temporal dimension allowing for flexible temporal resolution.
- An ablation and a comparison studies showing significant improvement w.r.t. existing motion priors used in a markerless capture context, against all sparsification factors in space and time.

2. Related Work

Methods that output a temporally coherent 4D human motion sequence based on untracked 2D or 3D input have been studied in computer vision and computer graphics in the past decades, mainly under the name of markerless performance capture, and sometimes *motion infilling* [36], or *motion interpolation* [18]. As it is ill-posed, different strategies have been proposed.

Template-based methods. A first line of work deforms a subject-specific template over time to fit it to observations. These methods have been applied to multi-view *e.g.* [32, 6, 34, 30], depth camera *e.g.* [17, 40] and monocular *e.g.* [37, 9] inputs in the past 15 years. As the template may include information on clothing, many of these methods take into consideration clothing and some even run in real-time. These methods allow for impressive results when subject-specific templates are available as input in addition to 2D or 3D videos.

Static and dynamic priors. Different data-driven strategies have been proposed to address the performance capture problem. They include leveraging statistical models of the human body, which are fitted to 2D or 3D video *e.g.* [1, 8, 19, 4, 13] and learning information on human body configurations [10]. These works have been extended to include information on faces and hands [31, 14, 23], clothing [38, 2, 20], dynamics [25, 3, 15, 16], and transitions between static frames [27]. Some methods output fully animatable avatars in addition to coherent 4D performance capture *e.g.* [29, 5]. These methods provide stable and accurate results for input data that is densely sampled in time. We demonstrate experimentally that our method outperforms a baseline that fits a state of the art static model [23] to the input frames and adds frames by interpolating the resulting fitted frames using a skeleton-based method.

Motion priors. To process data that is sampled sparsely in time, a recent line of work explores the use of motion priors where a full motion is represented in a low-dimensional space. Motion priors based on implicit spatio-temporal representations *e.g.* [22, 28] offer the advantage of being very flexible w.r.t. the spatial and temporal sampling of the input data. These works

have been specialized to human motions [11, 12]. Neural radiance field strategies have also been extended to human performance capture [35] and achieve impressive results when one or multiple 2D videos are available as input. Closer to our work, another strategy to learn human motion priors using parametric human body models has emerged [7, 24, 18, 36]. The first two methods consider different problems but contain ideas that inspired our method. Ghorbani *et al.* [7] synthesize motion variations, and subdivide long motions into smaller segments. Petrovich *et al.* [24] synthesize motions of a specific given action, and explicitly model the duration of a motion. Closest to our work, [18, 36] learn a motion prior from various motions performed by different actors and leverage them for human performance capture. We demonstrate experimentally that our method achieves higher precision than these works for markerless motion capture from point clouds.

3. Overview

We reconstruct a coherent 4D representation from an untracked and sparse sequence of 3D point clouds with a data-driven approach that learns a 4D human motion prior. As output and for training, our method represents coherent 4D motion sequences using a parametric human body model.

As our goal is to handle motions of varying duration and resolution, we learn our latent motion prior using seq2seq encoders [33], thereby representing a 4D motion sequence of arbitrary length and framerate by a sequence of latent motion primitives and an additional body shape parameter. Using a sequence of latent primitives instead of a single latent vector allows for better generalization to different motions because the 4D motion sequence is segmented and different segments are represented by different latent primitives. The segmentation is modeled using temporal masks that are differentiable, which allows to learn the segmentation without supervision.

To generate coherent output motions at arbitrary temporal resolution, we decode the sequence of latent primitives in a temporally implicit way to reconstruct a parametric human body model at any given time instant. This allows to reconstruct densely sampled coherent 4D motions from sparsely sampled inputs.

For training, we use a dataset of human motion sequences parameterized by a parametric human body model, for which we also generate corresponding untracked 3D point clouds. To learn the motion prior, we train a seq2seq encoder and temporally implicit decoder as a variational autoencoder (VAE). For training, we take as input parametric representations of the motion sequences to leverage the tracked models. This learns a motion prior that automatically segments the input sequence and represents it using a sequence of latent primitives. As we do not have access to coherent 4D motion sequences at inference time, we train a second seq2seq encoder that maps sequences of untracked 3D point clouds to the (already learned) motion prior.

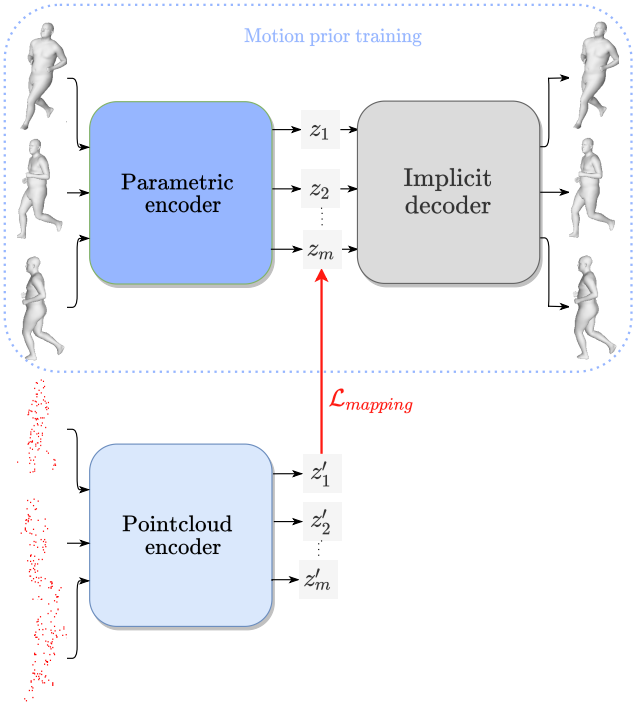

Figure 1. Method overview. Architecture consists of two seq2seq encoders (blue blocks) that map a human motion sequence into a sequence of latent primitives $z_1, \ldots, z_m$, and an implicit decoder (grey block) that decodes $z_1, \ldots, z_m$ into a parametric human body model. Training first learns a motion prior of latent primitives by leveraging parametric body models (top row), and subsequently learns a mapping $\mathcal{L}_{mapping}$ from untracked point cloud sequences to the resulting latent space (bottom row). At inference time, the point cloud encoder followed by $\mathcal{L}_{mapping}$ and the implicit decoder allow for robust initialization.

Encoding an untracked sequence of point clouds into a sequence of latent primitives, and decoding this sequence at arbitrary temporal resolution with a mean body shape allows for good initialization of the performance capture without tracked markers at inference time. This initialization can be further improved by optimizing for the latent primitives and body shape parameters such that the decoded meshes are close to the untracked input point clouds.

4. Method

Our method leverages a set of coherent 4D sequences for training, which are sampled sparsely in space and time to generate corresponding sequences of untracked 3D point clouds. This section outlines how coherent sequences are represented, details the different parts of our architecture, shown in Figure 1, and provides details on training and inference.

4.1. Coherent 4D sequence representation

We represent a temporally coherent 4D human motion sequence in a low-dimensional space with the help of a static parametric body model provided at every frame. While any model can be used, we base our implementation on the SMPL [19] model because the AMASS dataset [21] readily provides this parameterization. SMPL parameterizes a static 3D body model using body shape β, a skeleton pose θ and a displacement Γ. For skeleton pose, we use a subset of joints of the SMPL skeleton and to represent the trajectory in space, we use a 3D displacement vector.

As body shape is fixed over time, this allows to represent a motion sequence of duration d using β, $\theta(t)$, and $\Gamma(t)$, where $t \in [0, d]$ is the parameter controlling time. We call $M(t) = SMPL(\theta(t), \Gamma(t), \beta)$ the function that outputs the SMPL mesh corresponding to the parametric representation $\theta(t), \Gamma(t), \beta$. A discretized motion sequence consisting of n frames is represented by $\{\theta(t_i), \Gamma(t_i), \beta, t_i\}_{i=1}^{n}$, where t_i are the time stamps corresponding to the meshes. Furthermore, we denote a sequence of n untracked 3D point clouds by $\{P_i, t_i\}_{i=1}^{n}$ in the following. Note that 3D point clouds corresponding to $\{\theta(t_i), \Gamma(t_i), \beta, t_i\}$ can be obtained by sampling points from $M(t_i)$. In practice, we uniformly sample 100 points to generate training sequences.

4.2. Transformer encoders

We represent a 4D human motion sequence using two independent factors: a sequence of m latent primitives $\{z_i\}_{i=0}^{m}$, and body shape β. To allow for flexible duration and frame rates, we allow an arbitrary number of input frames per sequence. The latent primitives are obtained by a transformer encoder that maps an input sequence to a sequence of latent primitives. The transformer block of the encoder operates on a reduced representation (embedding) of the input frames and is similar to the original transformer [33] that was used for language translation. The difference between language translation and our setting is that we do not have ground truth for the latent primitives. To make learning this part in an unsupervised way feasible, we fix the number m of latent primitives. To allow for flexibility,

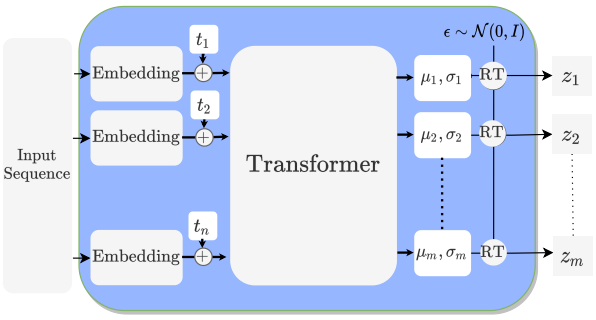

Figure 2. Encoder mapping an input sequence into a sequence of latent motion primitives $z1, \ldots, z_m$. The embedding is one multi layer perceptron for the parametric input and a PointNet [26] for point cloud input. Time stamps t_i are added as positional encoding. Transformer outputs a sequence latent distributions μ_i, σ_i from which z_i are sampled using the reparametrization trick (RT).

the duration d_i of each motion primitive is flexible and learned using a differentiable cost function.

The output sequences of the transformer encoder are interpreted as a sequence of Gausssian distributions defined by a sequence of means and standard deviations $\{\mu_i, \sigma_i\}_{i=1}^{m}$ from which the latent primitives $\{z_i\}_{i=1}^{m}$ are sampled using Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ such that $z_i = \mu_i + \epsilon\sigma_i$. This is similar to the interpretation of latent spaces of VAEs and known to allow for generalization.

To enable inference from untracked sequences of 3D point clouds while taking advantage of temporally coherent 4D sequences during training, our method uses two transformer encoders. The architectures of these encoders differ in the way the embedding of the input frames is obtained, and is shown in Figure 2.

Parametric encoder The parametric encoder considers $\{\theta(t_i), \Gamma(t_i), t_i\}_{i=1}^{n}$ parameterized by SMPL. To reduce the dimensionality of this representation, for each frame $\theta(t_i), \Gamma(t_i)$ are passed through one perceptron layer, and t_i is subsequently concatenated to this representation. Note that the perceptron applied to all frames share weights.

Point cloud encoder The point cloud encoder considers untracked point clouds $\{P_i, t_i\}_{i=1}^{n}$. To reduce the dimensionality in this case, each frame P_i is passed through a PointNet [26], and t_i is subsequently concatenated to this representation. All PointNets share weights.

4.3. Temporally implicit decoder

Our decoder operates in two stages by first decoding individual latent primitives and by subsequently combining them. This allows learning a motion prior on the latent primitives which each characterize a mo-

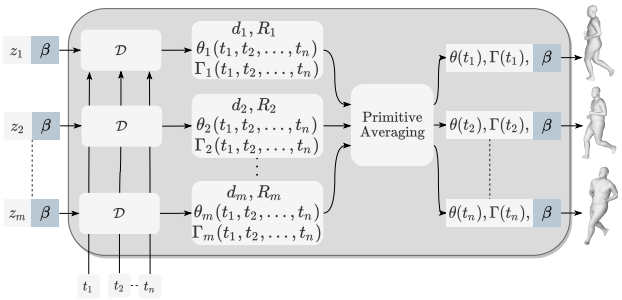

Figure 3. Implicit decoder. Given a sequence of latent primitives z_i, a body shape β and timestamps t_j, the decoder outputs a sequence of body meshes parameterized by β, θ, Γ. z_i are decoded independently into segment parameters that include duration d_i and transformation R_i, and subsequently combined to decode a dense 4D motion.

tion segment, while still enforcing coherence between neighboring segments. The architecture of the implicit decoder is shown in Figure 3.

Primitive decoding First, given a body shape β, the latent primitives are decoded individually using an implicit primitive decoder $\mathcal{D}(z_i, \beta, t)$ that outputs per-segment motion parameterized by $\theta_i(t), \Gamma_i(t)$, its duration d_i and a rigid transformation R_i. $\theta_i(t)$ and $\Gamma_i(t)$ characterize the global motion on the temporal segment $\tau_i, \tau_i + d_i$, where $\tau_i = \sum_{j<i} d_j$. This ensures that each latent primitive encodes information of a continuous temporal segment of the input motion. For invariance w.r.t. the initial orientation and displacement of a segment, $\mathcal{D}$ also outputs a rigid transformation R_i which is used as a transition matrix from segment space to input space. The architecture of $\mathcal{D}$ consists of two MLPs. The first MLP outputs $\theta_i(t)$,$\Gamma_i(t)$ as function of z_i, β and the segment time $t - \tau_i$. The second MLP outputs the per segment parameters R_i and d_i as a function of z_i and β.

Primitive combination To combine the segment representations into motion $\theta(t), \Gamma(t)$, a weighted average of the per-segment representations using temporal masks is computed. For each primitive, the corresponding Gaussian mask is $G_i(t) = e^{-\left(\frac{t - \frac{(\tau_i + d_i)}{2}}{d_i/2}\right)^2}$ such as:

$$\theta(t) = \frac{\sum_i G_i(t) R_i \theta_i(t)}{\sum_i G_i(t)}, \Gamma(t) = \frac{\sum_i G_i(t) R_i \Gamma_i(t)}{\sum_i G_i(t)}. \quad (1)$$

The temporally implicit nature of $\mathcal{D}$ alleviates the problem of averaging segments that may not be temporally aligned according to a predefined frame rate. The averaging of rotations is done in 6D representation space [39]. This qualitatively leads to naturally combined results.

4.4. Training

The training is divided into two stages. The first stage benefits from parametric training data to learn a motion prior. The second stage learns a mapping function from sequences of untracked point clouds to the prior learned in the first phase, thereby allowing for untracked input at inference time.

Motion prior In the first stage, the parametric encoder and the implicit decoder are trained in a self-supervised manner with a reconstruction loss, a Kullback–Leibler (KL) divergence loss to constrain the prior distribution to a normal distribution, and a regularization loss on the segment duration for faster convergence and to prevent local minima. The loss during this first stage is $\mathcal{L} = \mathcal{L}_{rec} + \lambda_{KL}\mathcal{L}_{KL} + \lambda_{reg}\mathcal{L}_{reg}$ where,

$$\begin{aligned} \mathcal{L}_{rec} &= \mathcal{L}_{global} + \mathcal{L}_{segment}, \\ \mathcal{L}_{KL} &= \tfrac{1}{m} \textstyle\sum_{j=1}^{m} KL(\mathcal{N}(\mu_j, \sigma_j), \mathcal{N}(0, I)), \\ \mathcal{L}_{reg} &= \textstyle\sum_{j=1}^{m} (d_j - 1/m)^2. \end{aligned} \quad (2)$$

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

$$\begin{aligned} \mathcal{L}_{global} = \frac{1}{n} \sum_{i=1}^{n} \Big[& \Big((\theta, \Gamma)(t_i) - (\theta_{GT}, \Gamma_{GT})(t_i) \Big)^2 \\ + \; & \lambda_{3D} \Big((M(t_i) - M_{GT}(t_i) \Big)^2 \Big] \end{aligned} \quad (3)$$

The second term $\mathcal{L}_{segment}$ acts as a per segment reconstruction loss, which guarantees that each segment represents a realistic motion and allows for realistic reconstructions where segments are overlapping.

$$\begin{aligned} \mathcal{L}_{segment} = \frac{1}{mn} \sum_{i=1}^{n} \sum_{j=1}^{m} G_j(t) \Big(& R_j(\theta_j, \Gamma_j)(t_i) \\ & - (\theta_{GT}, \Gamma_{GT}) \quad (t_i) \Big)^2 \end{aligned} \quad (4)$$

Mapping sequences of point clouds In a second stage, the parametric encoder and the implicit decoder weights are fixed and the point cloud encoder is trained as a mapping function from point cloud sequences to the latent primitives. In this stage, the loss has a single term $\mathcal{L}_{mapping}$ which is a distance between the sequence of primitives obtained from the point cloud encoder, $\{\mu'_j, \sigma'_j\}_{j=0}^{m}$, and the sequence of primitives obtained via the parametric encoder, $\{\mu_j, \sigma_j\}_{j=0}^{m}$:

$$\mathcal{L}_{mapping} = \frac{1}{m} \sum_{j=1}^{m} KL(\mathcal{N}(\mu_j, \sigma_j), \mathcal{N}(\mu'_j, \sigma'_j)) \quad (5)$$

4.5. Inference

At inference time, the input is a possibly sparsely sampled sequence of point clouds $\{P_i, t_i\}$ and our goal is to find parameters $z = \{z_i\}_{i=1}^m, \beta$ that when decoded represent P_i well. Inference proceeds in two steps: a direct inference used for initialization and an optimization to refine the result.

Initialization In a first step, $\{P_i, t_i\}$ is mapped to a latent representation z^{init} using the point cloud encoder and we initialize $\beta^{init} = 0$. This allows to have an initialization of the motion sequence, which can be decoded into a coherent representation at arbitrary frame rate using the implicit decoder.

Optimization To refine the result, we optimize for latent codes z^* and a body shape β^* using the differentiable decoder. The optimal parameters are obtained by minimizing the average Chamfer distance (CD) between the reconstructed meshes $\{M_{z,\beta}(t_i)\}$ and the ground truth point cloud sequence $\{P_i, t_i\}$. Given a sequence of n point clouds:

$$z^*, \beta^* = \underset{z,\beta}{argmin}\Big(\sum_i CD(M_{z,\beta}(t_i), P_i)\Big) \qquad (6)$$

5. Experimental results

We start by outlining the implementation and data used to build our model, evaluate the motion prior and present results of our method. More qualitative results are provided in supplementary material.

5.1. Implementation and data

Implementation details Our method is implemented using pytorch and the Adam optimizer is used for optimization. SMPL represents a static body by 22 skeleton joints and 16 body shape parameters. We discard the foot joints which have constant rotation in AMASS, which results in 20 skeleton joints. Each joint is represented in 6D using its relative rotation to its parent joint [39]. For body shape, we use the first 8 shape components. 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 from the training set. During training, global displacements are normalized to $[-1, 1]$ for each direction and timestamps are scaled to $[0, 1]$. Unless stated otherwise, we set $m = 8$ and each latent vector has dimension $D = 256$. Additional details are given in supplemental material.

Training data For training, we use a subset of AMASS [21], a collection of different datasets parameterized by SMPL that contains a variety of motions and body shapes. We leave part of this dataset for validation by removing full datasets.

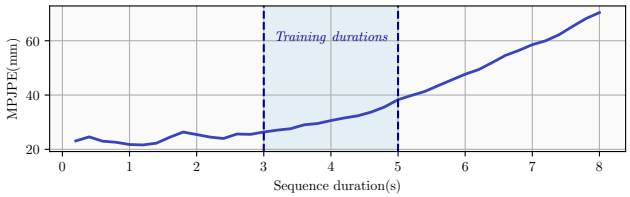

Figure 4. Generalization to sequence duration outside training set (for training, duration of $3 - -5s$). Blue lines show evolution of MPJPE (lower is better) of our method for different sequence duration.

Test data For testing, we use two datasets. The first one is the left out part of AMASS that is used to evaluate the motion prior. As all data in AMASS was created by fitting SMPL to MoCap data, we perform our comparative evaluations on a different dataset to avoid overfitting to a specific data creation setup. This second dataset consists of sequences of 3D point clouds (approximately 10000 points per frame) captured using a multi-view camera system at 50 frames per second and no parametric representation is available, called *multi-view test set* in the following. The data consists of 4 subjects (2 males, 2 females) performing different motions including boxing, kicking, sidestepping and various types of walking and running. In total, 170 motion sequences are used for testing. This test set allows to evaluate the robustness to degraded signals by progressively downsampling the spatial and temporal resolution of the input data.

5.2. Evaluation of the motion prior

To evaluate the motion prior, we test the generalization of the to sequences of duration outside of the training set, evaluate the influence of the sequential latent representation, and the segmentation learning. All of the evaluations use the AMASS test set.

Generalization Our model is learned using sequences with duration $3 - -5s$ and $m = 8$ latent primitives. We analyze how well this model generalizes to sequences of different duration by applying it to duration $0.2 - -8s$. To process sequences longer than $5s$ with our method, they are virtually accelerated by scaling the timestamps to $[0, 1]$. We encode test sequences of different duration using the parametric encoder, decode them using the implicit decoder and consider the mean per joint position error (MPJPE) between input and output joints. This error is averaged over the sequence. Figure 4 shows the evolution of MPJPE for different sequence duration. Our model generalizes well to sequences of different duration, especially for sequences that are shorter than those used during training.

Sequential latent representation To evaluate

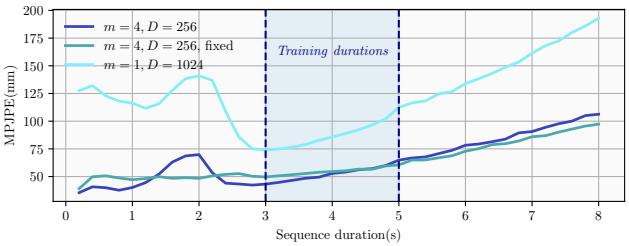

Figure 5. Value of using sequences of latent primitives and flexible segmentation. Lines show evolution of MPJPE (lower is better) of our method for different sequence duration ($3-5s$ sequences used for training). Note that using latent sequences with flexible segmentation $m = 4, D = 256$ performs best in training interval and generalizes well.

the value of learning a sequence of latent primitives instead of a single latent vector, we compare our method to the result when setting $m = 1$. To compare the same total number of latent dimensions, we consider two models in this experiment: one trained with $m = 4$ where each segment has $D = 256$ latent dimensions and one with $m = 1$ and $D = 1024$. For evaluation, we consider the same generalization plot to sequences of different duration as before. Note that m is reduced in this setting for practical reasons as training for $m = 1$ with high latent dimension is costly. Figure 5 shows that $m = 4, D = 256$ leads to significantly lower MPJPE than $m = 1, D = 1024$ and generalizes significantly better to sequences of different duration. This shows that allowing for sequential latent space primitives better represents varying motions and duration.

Segmentation learning To evaluate the influence of flexible segments duration learned in an unsupervised way, we compare our method to one trained with fixed segmentation parameters $d_i = 1/m$. As in the previous experiment, we set $m = 4, D = 256$ for both models and consider the same generalization plot to sequences of different duration. Figure 5 shows that the differences between the two models are minor. Allowing for flexible segments slightly degrades performance for longer sequences; the virtual acceleration of sequences is more detrimental to the model when learning the segmentation parameters as they are more heavily influenced by timestamp variations. However, flexible segments slightly improve the model performance in the training interval.

Influence of optimization The initialization of the latent motion primitives provided by the point cloud encoder is not precise enough to directly reconstruct dense 4D motion. To quantify the influence of optimization, we take a set of $5s$ motions from the AMASS test set and sample 1000 points from each frame. Using the point cloud encoder, we obtain an initialization z^{init} for each sequence, which in combination with $\beta^{init} = 0$, leads to an average per vertex error of $140mm$. By optimizing for z^* and β^*, the average error improves significantly to $33mm$.

5.3. Comparative evaluation

We now provide comparative evaluations of our method w.r.t. a strong baseline based on a static parametric human body model and recent state of the art motion priors. As most existing methods cannot process sequences of arbitrary duration, our evaluation focuses on how well the methods perform when the input signals are degraded, by considering inputs at increasingly sparse spatial and temporal resolutions.

Evaluation protocol To prevent any biases from having learned on AMASS, all comparative evaluations are performed on the multi-view test set. For this dataset, we have sequences at 50 fps where each frame contains approximately 10000 points. To evaluate the robustness of the methods w.r.t. degraded input signals, we downsample the input sequences spatially to 100 and 1000 points per frame, and temporally to 5 and 10 fps. For each experiment, we reconstruct coherent 4D sequences at 30 fps which is the proposed framerate in [36] and [18] and evaluate the error of the markerless performance capture by the 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 $4s$. All comparisons are based on code and pre-trained models provided with the respective publications.

Parametric baseline VPoser+SLERP The first baseline relies on the state of the art static pose prior VPoser [23]. In this approach, a latent pose representation is output per frame. As the global displacement is not encoded in VPoser, we optimize additionally a displacement per input frame. To increase the temporal resolution, we linearly interpolate between observed frames for displacement and using spherical linear interpolations (SLERP) for body pose rotations. We call this baseline VPoser+SLERP in the following.

Motion prior with frequency guidance [36] We compare to two parametric human motion priors. The first one uses frequency guidance and was trained for motions of fixed duration ($4s$) at 30 fps. This prior does not encode global displacements, and we optimize them additionally for input frames and interpolate linearly for the remaining frames.

Hierarchical motion prior [18] The second parametric human motion prior uses a hierarchical approach to encode motions of fixed duration ($2s$) at 30 fps. As we consider sequences of $4s$ in our comparisons,

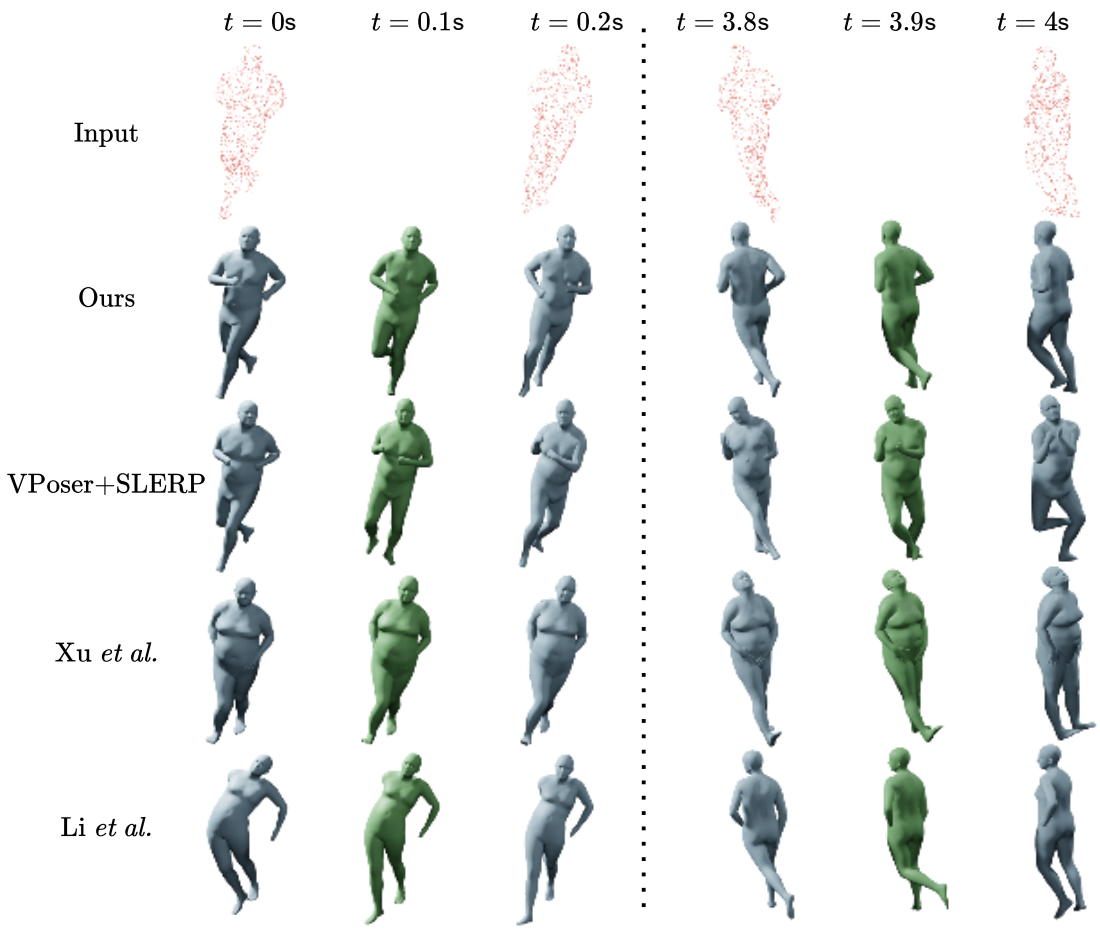

Figure 6. Comparison to state of the art on a challenging example. Completion results on a sequence of a person running in circle; we show frames close to the beginning and the end of the sequence. Our method estimates pose more precisely than other strategies. Blue meshes approximate input frames, green meshes are interpolated by the motion priors.

we optimize for two segments for this method.

Results and discussion Table 1 reports the results obtained when completing from 5 and 10fps motion to a 30fps motion, when considering input sequences increasingly sparsely sampled in space and time. Note that our method degrades gracefully for decreasing input resolutions. Our method significantly outperforms state of the art methods. These improvements are due to the good initialization provided by our point cloud encoder, which prevents the optimization from getting stuck in local minima due to the symmetry of the human body. Additionally, we capture more details than the other two parametric motion priors due to our sequential representation of the motion. Figure 6 shows an example where other methods fail due to the global orientation, while our method achieves plausible results thanks to the initialization by the point cloud encoder. Xu *et al.* and VPoser+SLERP do not encode global displacement and fail to reconstruct the correct orientation of the last frames. While Li *et al.* encode global displacement, our pose estimation is more plausible for the first frames.

# points per frame	100		1000		10000	
Input fps	5	10	5	10	5	10
VPoser+SLERP	27	25	24	20	24	20
Xu *et al.* [36]	28	26	26	24	26	24
Li *et al.* [18]	83	79	52	48	42	40
Ours	**21**	**17**	**17**	**13**	**17**	**13**

Table 1. Comparison to state of the art using average Chamfer distance (mm) (lower is better). Markerless performance capture with input sequences of different spatial (# points) and temporal (fps) resolutions.

6. Conclusion

This work presented a spatio-temporal representation of motion as a sequence of latent primitives for the

task of markerless performance capture from sequences of untracked 3D point clouds. It showed that using latent primitives characterizing temporal segments allows for a gain in precision which outweighs the gain of adding more latent dimensions. We also showed that using motion priors for markerless performance capture from untracked data requires a good latent initialization and proposed a solution by training an additional point cloud sequence encoder as a mapping function. We demonstrated that when considering increasingly sparsely sampled input data, our method maintains high precision compared to state of the art approaches.

7. Acknowledgements

We thank Diego Thomas and Raphaël Dang Nhu for interesting discussions about the method design and paper redaction. We also thank Julien Pansiot and Laurence Boissieux for managing the acquisition and data processing of the multi-view test set. 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.

References

[1] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: shape completion and animation of people. *Transactions on Graphics*, 24(3):408–416, 2005. 2

[2] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-Garment Net: learning to dress 3d people from images. In *International Conference on Computer Vision*, 2019. 2

[3] Federica Bogo, Michael J Black, Matthew Loper, and Javier Romero. Detailed full-body reconstructions of moving people from monocular RGB-D sequences. In *International Conference on Computer Vision*, 2015. 1, 2

[4] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter V. Gehler, Javier Romero, and Michael J. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision*, 2016. 2

[5] Andrei Burov, Matthias Nießner, and Justus Thies. Dynamic surface function networks for clothed human bodies. In *International Conference on Computer Vision*, 2021. 2

[6] Edilson De Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. *Transactions on Graphics*, pages 1–10, 2008. 2

[7] Saeed Ghorbani, Calden Wloka, Ali Etemad, Marcus A Brubaker, and Nikolaus F Troje. Probabilistic character motion synthesis using a hierarchical deep latent variable model. In *Computer Graphics Forum*, volume 39, pages 225–239, 2020. 3

[8] Peng Guan, Alexander Weiss, Alexandru O Balan, and Michael J Black. Estimating human shape and pose from a single image. In *International Conference on Computer Vision*, 2009. 2

[9] Marc Habermann, Weipeng Xu, Michael Zollhöfer, Gerard Pons-Moll, and Christian Theobalt. Livecap: Real-time human performance capture from monocular video. *Transactions on Graphics*, 38(2), 2019. 2

[10] Zeng Huang, Tianye Li, Weikai Chen, Yajie Zhao, Jun Xing, Chloe LeGendre, Linjie Luo, Chongyang Ma, and Hao Li. Deep volumetric video from very sparse multi-view performance capture. In *European Conference on Computer Vision*, 2018. 2

[11] Boyan Jiang, Yinda Zhang, Xingkui Wei, Xiangyang Xue, and Yanwei Fu. Learning compositional representation for 4d captures with neural ODE. In *Conference on Computer Vision and Pattern Recognition*, 2021. 3

[12] Boyan Jiang, Yinda Zhang, Xingkui Wei, Xiangyang Xue, and Yanwei Fu. H4D: human 4d modeling by learning neural compositional representation. In *Conference on Computer Vision and Pattern Recognition*, 2022. 3

[13] Haiyong Jiang, Jianfei Cai, and Jianmin Zheng. Skeleton-aware 3d human shape reconstruction from point clouds. In *International Conference on Computer Vision*, 2019. 1, 2

[14] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *Conference on Computer Vision and Pattern Recognition*, 2018. 2

[15] Meekyoung Kim, Gerard Pons-Moll, Sergi Pujades, Seungbae Bang, Jinwook Kim, Michael J Black, and Sung-Hee Lee. Data-driven physics for human soft tissue animation. *Transactions on Graphics*, 36(4):1–12, 2017. 2

[16] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2

[17] Hao Li, Bart Adams, Leonidas J. Guibas, and Mark Pauly. Robust single-view geometry and motion reconstruction. *Transactions on Graphics*, 28(5):1–10, 2009. 2

[18] Jiaman Li, Ruben Villegas, Duygu Ceylan, Jimei Yang, Zhengfei Kuang, Hao Li, and Yajie Zhao. Task-generic hierarchical human motion prior using vaes. *Conference on 3D Vision*, 2021. 1, 2, 3, 7, 8

[19] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: a skinned multi-person linear model. *Transactions on Graphics*, 34(6):1–16, 2015. 2, 4

[20] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J.

Black. Learning to dress 3d people in generative clothing. In *Conference on Computer Vision and Pattern Recognition*, June 2020. 2

[21] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, 2019. 4, 6

[22] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *International Conference on Computer Vision*, 2019. 2

[23] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 7

[24] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer VAE. In *International Conference on Computer Vision*, 2021. 3

[25] Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J. Black. DYNA: a model of dynamic human shape in motion. *Transactions on Graphics*, 34:120:1–120:14, 2015. 2

[26] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: deep learning on point sets for 3d classification and segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2017. 4

[27] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. Humor: 3d human motion model for robust pose estimation. In *International Conference on Computer Vision*, 2021. 2

[28] Davis Rempe, Tolga Birdal, Yongheng Zhao, Zan Gojcic, Srinath Sridhar, and Leonidas J Guibas. CASPR: learning canonical spatiotemporal point cloud representations. *Advances in neural information processing systems*, 33:13688–13701, 2020. 2

[29] Helge Rhodin, Nadia Robertini, Dan Casas, Christian Richardt, Hans-Peter Seidel, and Christian Theobalt. General automatic human shape and motion capture using volumetric contour cues. In *European Conference on Computer Vision*, 2016. 1, 2

[30] Nadia Robertini, Dan Casas, Edilson De Aguiar, and Christian Theobalt. Multi-view performance capture of surface details. *International Journal on Computer Vision*, 124(1):96–113, 2017. 2

[31] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *Transactions on Graphics*, 36(6), Nov. 2017. 2

[32] Jonathan Starck and Adrian Hilton. Surface capture for performance-based animation. *Computer Graphics and Applications*, 27(3):21–31, 2007. 2

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 3, 4

[34] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. Articulated mesh animation from multi-view silhouettes. *Transactions on Graphics*, pages 1–9, 2008. 2

[35] Hongyi Xu, Thiemo Alldieck, and Cristian Sminchisescu. H-NeRF: neural radiance fields for rendering and temporal reconstruction of humans in motion. *Advances in Neural Information Processing Systems*, 34, 2021. 3

[36] Jiachen Xu, Min Wang, Jingyu Gong, Wentao Liu, Chen Qian, Yuan Xie, and Lizhuang Ma. Exploring versatile prior for human motion via motion frequency guidance. In *Conference on 3D Vision*, 2021. 1, 2, 3, 7, 8

[37] Weipeng Xu, Avishek Chatterjee, Michael Zollhöfer, Helge Rhodin, Dushyant Mehta, Hans-Peter Seidel, and Christian Theobalt. Monoperfcap: Human performance capture from monocular video. *Transactions on Graphics*, 37(2):1–15, 2018. 1, 2

[38] Tao Yu, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Qionghai Dai, Hao Li, Gerard Pons-Moll, and Yebin Liu. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *Conference on Computer Vision and Pattern Recognition*, 2018. 2

[39] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Conference on Computer Vision and Pattern Recognition*, 2019. 5, 6

[40] Michael Zollhöfer, Matthias Nießner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, and Marc Stamminger. Real-time non-rigid reconstruction using an RGB-D camera. *Transactions on Graphics*, 33(4):1–12, 2014. 2