Controllable Variation Synthesis for Surface Motion Capture

Adnane Boukhayma, Edmond Boyer

Controllable Variation Synthesis for Surface Motion Capture

Adnane Boukhayma, Edmond boyer
Inria, LJK, Univ. Grenoble Alpes
firstname.lastname@inria.fr

Abstract

We address the problem of generating variations of captured 4D models automatically, and we particularly focus on dynamic human shapes as observed from multi-view videos. Variation is an essential component of motion realism, however recent mesh animation datasets and tools lack such richness. Given a few 4D models representing movements of the same type, our method builds a probabilistic low dimensional embedding of shape poses using Gaussian Process Dynamical Models, and novel variants of motions are obtained by sampling trajectories from this manifold using Monte Carlo Markov Chain. We can synthesise an unlimited number of variations of any of the input movements, and also any blended version of them, without costly non-linear interpolation of input movement variations in mesh domain. The output variations are statistically similar to the input movements but yet slightly different in poses and timings. As we show through our results, the generated mesh sequences match the training examples in realism, which facilitates 4D model dataset augmentation.

1. Introduction

4D surface capture is an emerging tool for acquiring 3D dynamic shape models of moving objects either with high quality multi-view set-ups [10, 11, 32, 31, 33] or with affordable low-cost sensors [37, 26, 17]. Such data, which combines shape and kinematic information of the object of interest, can find many applications such as 3D content generation, augmenting machine learning training sets or statistical modelling. Surface tracking [1, 24, 6], appearance modelling [34, 5], animation synthesis [9, 3, 28, 4] and other tasks related to the acquisition and reuse of 4D data have received increasing interest from the vision and graphics communities lately. Like many of these works, we focus on human corpus as observed from multi-view videos.

As humans rarely perform similar actions in the exact same manner every time, variation in motion is an essential component of animation realism. Unfortunately, current 4D data animation solutions [9, 3, 28] merely replay input motion segments or blended versions of them. Such exact repetition of motion cycles can lead to unrealistic animations. Hence, a variation model that can generate even slight differences of the original motion cycles can improve the naturalness of the generated animations substantially, and provide new examples without the burden of motion acquisition.

In this work, we address the task of generating an unlimited number of variations of a subject movement using a limited number of training frames of captured performance. Given a few examples of a particular type of motion as input, such as locomotion (walking, running, turning, etc.) or jumping (jumping far, close, high, low, etc.), a manifold of motion is learned and new variations are obtained by sampling from the later. The user can in particular generate variations of any of the input motions and also any blended version of them. While this problem has received some attention previously for sparse surface representations, namely traditional Motion Capture [36, 19, 21], to our knowledge this is the first work of its type that considers dense surface representations relevant to our capture situation.

Our main contributions in this work are as follows: First, we advocate the use of Gaussian Process Dynamical Models directly on mesh data for the first time to build a low dimensional embedding of mesh based motion sequences. This two-fold model learns a probabilistic mapping between latent coordinates and mesh vertex coordinates, along with a second probabilistic mapping between successive frame latent coordinates. New motion examples are next generated by sampling from this manifold using a Hybrid Monte Carlo Markov Chain. The resulting sequences are statistically similar to the initialization of the Markov Chain but are not exact copies of it, as one could see in the latent space (figure 5) and the observation space (figures 8 and 9). This variation in the generated sequences stems from the following: variation in poses in the input data, the independent body part modelling scheme that we elaborate in section 4, the probabilistic mapping between the latent and the observation spaces, and finally the probabilistic mapping of dynamics in the latent space. Second, we propose an algorithm that allows generation of variations of any blended version

of the input sequences. This process avoids costly non-linear mesh interpolation of many variations in the observation space, and sampling motion around latent trajectories outside the training set, which leads to degenerate samples. Instead, we learn the model with few pre-interpolated sequences and sample variations only around learned trajectories. For a given requested blending weight, the variations of the closest latent trajectories are interpolated with the appropriate proportions to generate the requested blended sequence variations.

We evaluate our work perceptually in section 8 and in the accompanying video using a dataset of surface capture, with two different types of movements: locomotion and jumps. We succeeded in generating variations of the input motions and blended versions of them that are globally similar to the inputs but yet slightly different in both poses and timings. We also provide in section 7 numerical validation of the benefit of our motion parametrization scheme guided with pre-blended examples, compared to a simple trajectory interpolation in the latent space.

2. Related work

Variation versus noise With the term *variation* of motion, we refer to new examples that look globally like the original sequences but differ slightly from them in body poses and their timings of occurrence, thus mimicking human behaviour richness and inexactitude when reproducing the same movement. Previous work on Motion Capture data [2][27] attempts to generate variations merely with an additive noise component. However, biomechanical research [14] asserts that variation is rather a functional component of motion and not just noise. In addition, the work of [19] on motion capture shows empirically that there are no guarantees that added noise, either arbitrary or with tuned distributions, matches well with the existing motion, which renders these methods prone to unrealistic results and not robust to automation. Following other works on Motion Capture [36, 19, 21], we use a data-based approach where variation comes from the data and not a separate additive component.

Skeletal Motion Capture Many works use few examples of human Motion Capture sequences to generate new variations. The work of [29, 30] models the correlations between the degrees of freedom in motion data with a distribution, and synthesize new motions by sampling from this distribution. The authors in [19] use Dynamic Bayesian Networks to model and simulate dynamics in *similar but slightly different* example motions. [21] interpolates various examples of skeleton joint group motions with Universal Kriging. We follow the work of [36] and extend Gaussian Process Dynamical Models [35] to Surface Motion Capture. The GPDM augments the Gaussian Process Latent Variable Model [20, 13] with a latent dynamical model, in addition to a probabilistic non-linear mapping from the latent space to the data space. The dynamic model enables prediction and adds regularization when modelling temporal data, and was used successfully for standard Motion Capture data synthesis.

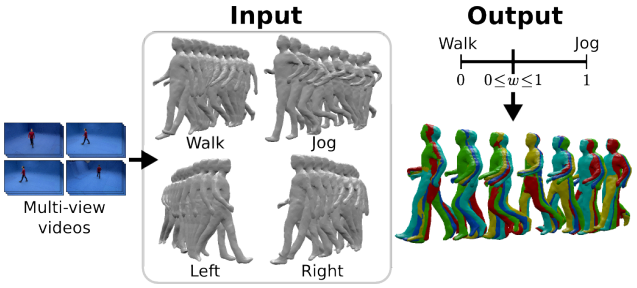


Figure 1: Overview. Left: Multi-view acquisition of motion sequences of the same type (e.g. locomotion). Right: Variation synthesis for any of the input sequences or any blended version of them. 5 variations are overlapped, each with a different color.

Mesh sequence parametrization Motion sequence parametrization [7, 18] is a key component in building parametric motion graphs [9, 15, 12] for interactive character control using skeletal and mesh data alike. For a given set of sequences exhibiting variation of the same type of movement, these sequences are temporally aligned and blended with various weights. However, each blending weight gives a unique output sequence. We extend this framework in our work to allow both motion blending and variation synthesis for blended motions in the latent motion space.

3. Method Overview

Our approach considers as input 3D shape sequences of the same subject as acquired from multi-view acquisition systems (See figure 1), performing motions of the same type, such as locomotion or jumping. Shapes are represented by globally consistent 3D meshes. In practice, motion sequences are temporally pre-aligned to a reference sequence and shapes are represented as independent body parts. We proceed as follows:

1. Motion variation: A Gaussian Process Dynamical Model is used to build a probabilistic low dimensional embedding from the input motions. We use a Hybrid Monte Carlo Markov Chain initialized with a latent motion trajectory to generate variants of this latter. Mesh motion sequences can then be obtained from latent trajectories through the Gaussian Process

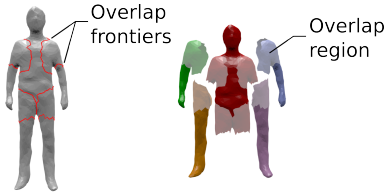


Figure 2: Coarse body partition.

of pose reconstruction. Variation in the generated outputs comes thus from: variation in poses in the input data, body part independent modelling, probabilistic mapping from latent to observation space, and probabilistic dynamic modelling.

2. Motion parametrization: for the sequences that the user wishes to parametrize, we learn the model with few non-linearly pre-interpolated intermediate sequences along with the original ones. We sample variations only around learned trajectories to avoid degenerate samples, and interpolate variations in the latent space of nearby latent trajectories, thus avoiding many non-linear mesh interpolations of variations in the mesh domain.

4. Shape motion representation

We represent shape in motion in the form of a 3D mesh $\mathcal{M}$ with consistent topology and connectivity. Shape motion can be characterized by elements such as sequential global rigid transformations and body part poses. While not fully independent, modelling limb and global body movements independently produces better results in our experiments. Consequently, we choose to learn motion for body elements using independent models.

Rigid alignment For every motion sequence of N frames: $\{\mathcal{M}_i\}_{1\leq i\leq N}$, we use orthogonal Procrustese analysis to align all frames to a template mesh. The motion sequence can hence be decomposed into a set of rigidly aligned meshes $\{\bar{\mathcal{M}}_i\}_{1\leq i\leq N}$ and relative rigid displacements between successive frames $\{\delta T_i, \delta R_i\}_{1\leq i\leq N}$, factored into elementary translations δT_i and rotations δR_i. These transformations are expressed with a 6-dimensional linear parametrization $(\delta T_i, \delta R_i) \mapsto (t_x, t_y, t_z, h_1, h_2, h_3)^T$, based on exponential maps for the rotation parameters h_k.

Body parts Each aligned mesh $\bar{\mathcal{M}}_i$ is next decomposed into P body part sub-meshes $\{\mathcal{P}_i^k\}_{1\leq k\leq P}$. We use $P = 5$ parts in a tree structured hierarchy including a torso as the root, and a pair of arms and legs as children nodes. We adopt the coarse and overlapping body segmentation strategy proposed in [4]. As illustrated in figure 2, the user provides closed and non-intersecting curves that delimit each overlapping region between contiguous body parts on $\mathcal{M}$. During motion learning and prediction, each body part is augmented with the overlap regions shared with its neighbouring parts. The sampled output body part motions are merged in the end with an automatic algorithm that ensures seamless body part stitching as described in section 6. Considering body parts independently allows for learning their deformations more accurately, enriches variation in the finally stitched body mesh, and also reduces the dimensionality of our model inputs. Increased complexity for large input partial meshes can be compensated by adopting more body parts with fewer vertices, or any means of dimension reduction such as PCA.

Each body part $\mathcal{P}_i^k$ is finally represented with a vector $\mathbf{y}_i$ that stacks successively the three coordinates of every vertex in the part. The torso vector is appended additionally with the 6 global mesh displacement parameters.

5. Shape motion embedding

We learn a probabilistic low dimensional embedding of motion for each body part independently using the Gaussian Process Dynamical Model [36]. We build latent spaces of body part motion using temporally pre-aligned motion sequences that are logically compatible, such as locomotion movements with various speeds and directions, or jumping movements with varying heights and lengths.

Non-linear dynamic model For a body part motion sequence $\{\mathbf{y}_i\}_{1\leq i\leq N}$, the GPDM comprises a non-linear probabilistic mapping f from the latent variables $\mathbf{x}_i \in \mathbb{R}^d$ to the meshes $\mathbf{y}_i \in \mathbb{R}^D$, parametrized with coefficients $\mathbf{A}$, and another mapping g between latent coordinates of consecutive frames, parametrized with coefficients $\mathbf{B}$:

$$\mathbf{y}_i = f(\mathbf{x}_i, \mathbf{A}) + \mathbf{n}_{y,i} \quad (1)$$

$$\mathbf{x}_i = g(\mathbf{x}_{i-1}, \mathbf{B}) + \mathbf{n}_{x,i} \quad (2)$$

where $\mathbf{n}_{x,i}$ and $\mathbf{n}_{y,i}$ are zero-mean white Gaussian noises.

Marginalizing over $\mathbf{A}$ [22, 25] with isotropic Gaussian priors on its parameters yields the likelihood:

$$P(\mathbf{Y}|\mathbf{X}, \boldsymbol{\alpha}) = \frac{1}{\sqrt{(2\pi)^{ND}|\mathbf{K}_Y|^D}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{K}_Y^{-1}\mathbf{Y}\mathbf{Y}^T\right)\right) \quad (3)$$

where $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^T$ is the matrix of training shape poses, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$ is the matrix of the associated latent positions. This likelihood is expressed using

the kernel matrix $\mathbf{K}_Y$ whose coefficients $(\mathbf{K}_Y)_{1\leq i,j\leq N} = k_Y(\mathbf{x}_i, \mathbf{x}_j)$ are defined using the Radial Basis Function [22]:

$$k_Y(\mathbf{x}, \mathbf{x}') = \alpha_1 \exp\left(-\frac{\alpha_2}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right) + \frac{\delta_{\mathbf{x},\mathbf{x}'}}{\alpha_3} \quad (4)$$

Hyperparameter vector $\boldsymbol{\alpha}$ comprises kernel parameters α_k and the variance of the additive noise $\mathbf{n}_{y,i}$.

Similarly, the density over latent trajectories can be obtained by marginalizing out $\mathbf{B}$ [35] with isotropic Gaussian priors on its parameters:

$$P(\mathbf{X}|\boldsymbol{\beta}) = \frac{P(\mathbf{x}_1)}{\sqrt{(2\pi)^{(N-1)d}|\mathbf{K}_X|^d}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{K}_X^{-1}\mathbf{X}'\mathbf{X}'^T\right)\right) \quad (5)$$

where $\mathbf{X}' = [\mathbf{x}_2, \ldots, \mathbf{x}_N]^T$ and $\mathbf{x}_1$ is given an isotropic Gaussian prior. This joint probability is expressed using the kernel matrix $\mathbf{K}_X$ whose coefficients $(\mathbf{K}_X)_{1\leq i,j\leq N-1} = k_X(\mathbf{x}_i, \mathbf{x}_j)$ are defined using the linear and Radial Basis kernel:

$$k_X(\mathbf{x}, \mathbf{x}') = \beta_1 \exp\left(-\frac{\beta_2}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right) + \beta_3 \mathbf{x}^T\mathbf{x}' + \frac{\delta_{\mathbf{x},\mathbf{x}'}}{\beta_4} \quad (6)$$

Hyperparameter vector $\boldsymbol{\beta}$ comprises kernel parameters β_k and the variance of the additive noise $\mathbf{n}_{x,i}$.

The latent variables $\mathbf{X}$ and the hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are estimated by minimizing the negative log of the posterior:

$$P(\mathbf{X}, \boldsymbol{\beta}, \boldsymbol{\alpha}|\mathbf{Y}) \propto P(\mathbf{Y}|\mathbf{X}, \boldsymbol{\alpha})P(\mathbf{X}|\boldsymbol{\beta})P(\boldsymbol{\alpha})P(\boldsymbol{\beta}) \quad (7)$$

Priors that favour small output scale are adopted for the hyperparameters: $P(\boldsymbol{\alpha}) = \Pi_k \alpha_k^{-1}$, $P(\boldsymbol{\beta}) = \Pi_k \beta_k^{-1}$. In our experiments, hyperparameters α_k and β_k are initialized with value 1, variances of noises $\mathbf{n}_{y,i}$ and $\mathbf{n}_{x,i}$ are initialized with value 0.1, and latent positions $\mathbf{X}$ are initialized with Principal Component Analysis coordinates. The latent space dimension is set to $d = 3$.

Multiple sequences The GPDM can naturally be extended to multiple motion sequences. After aligning them temporally to one reference sequence using Dynamic Time Warping [23], we concatenate the M training motion sequences $\mathbf{Y} = [\mathbf{Y}_1^T, \ldots, \mathbf{Y}_M^T]^T$. The associated latent position sequences $\{\mathbf{X}_j\}$ share the same latent space, $\mathbf{X}'$ comprises all but the first latent position for each sequence, and kernel matrix $\mathbf{K}_X$ is computed from all but the last latent position of each sequence. Figure 3 shows 4 latent trajectories of locomotion sequences of the right leg body part from DAN [9] dataset. Figure 4 shows vertex reconstruction errors of the training mesh sequences from two GPDMs built with walking and jumping movements from the same dataset, compared to PCA.

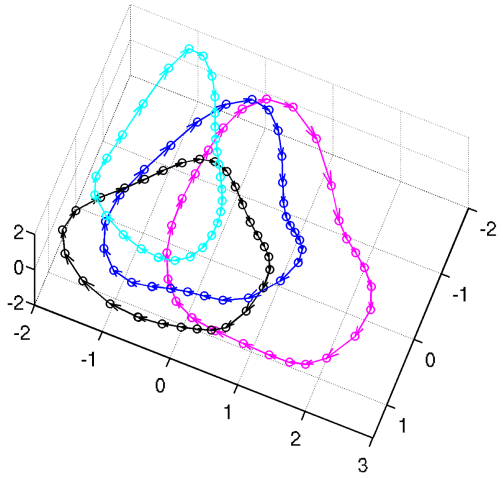


Figure 3: Latent trajectories of the right leg, learned with sequences *Walk* (blue), *Jog* (black), *Left* (cyan) and *Right* (magenta).

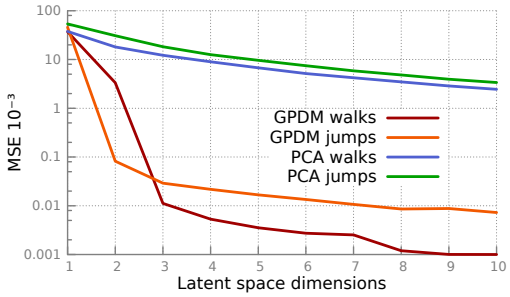


Figure 4: Mean reconstruction error of the *"walk"* GPDM (built with sequences *Walk*, *Jog*, *Left*, *Right*) and the *"jumps"* GPDM (built with sequences *Jump long*, *Short*, *High*, *Low*), compared to PCA.

6. Shape motion sampling

Following the work of [36] on skeletal Motion Capture data, we synthesise variants of a motion sequence $\mathbf{Y}_j$ in the observation space by sampling variants of its subjacent trajectory $\mathbf{X}_j = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ using a Markov Chain Monte Carlo method in the latent space, initialized with a mean prediction sequence. Mesh motion sequences are finally obtained from latent motion samples through the Gaussian Process of pose reconstruction.

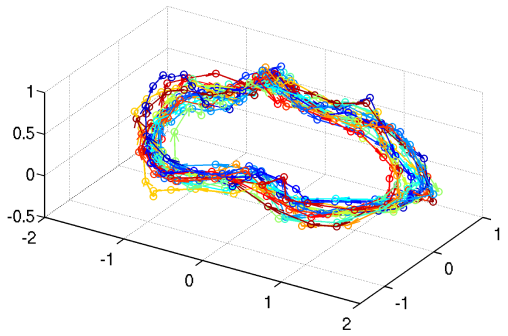


Figure 5: 20 latent variations of sequence *Walk* for the torso.

Mean prediction We start from the initial position $\mathbf{x}_1$ of the latent trajectory and simulate the dynamical process one frame at a time until we reach the original trajectory length N using mean prediction, that is the density over a latent prediction $\hat{\mathbf{x}}_i$ conditioned on the previous one $\hat{\mathbf{x}}_{i-1}$ is Gaussian [22]:

$$\hat{\mathbf{x}}_i \sim \mathcal{N}(\mu_X(\hat{\mathbf{x}}_{i-1}), \sigma_X^2(\hat{\mathbf{x}}_{i-1})\mathbf{I}) \quad (8)$$

$$\mu_X(\mathbf{x}) = \mathbf{X}'^T \mathbf{K}_X^{-1} \mathbf{k}_X(\mathbf{x}) \quad (9)$$

$$\sigma_X^2(\mathbf{x}) = \mathbf{k}_X(\mathbf{x},\mathbf{x}) - \mathbf{k}_X(\mathbf{x})^T \mathbf{K}_X^{-1} \mathbf{k}_X(\mathbf{x}) \quad (10)$$

where vector $\mathbf{k}_X(\mathbf{x})$ contains $k_X(\mathbf{x}, \mathbf{x}_i)$ in its i-th entry, and $\mathbf{x}_i$ is the i-th training vector.

Latent sampling Next, we use the resulting mean prediction sequence $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_n]^T$to initialize a Markov Chain that draws multiple fair samples of entire trajectories $\tilde{\mathbf{X}} \sim P(\hat{\mathbf{X}}|\mathbf{x_1}, \mathbf{X}, \mathbf{Y}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ offline using Hybrid Monte Carlo [22]. Figure 5 shows 20 variations, each with a different color, of the latent trajectory of the torso body part from sequence *Walk* of DAN dataset. The latent space was learned with Locomotion sequences *walk*, *Jog*, *Left* and *Right* of this very dataset.

Pose reconstruction Finally for a latent position $\tilde{\mathbf{x}}$ in a sampled trajectory $\tilde{\mathbf{X}}$, the corresponding shape $\tilde{\mathbf{y}}$ is obtained by sampling from the predictive distribution of the Gaussian Process of pose reconstruction [22]:

$$\tilde{\mathbf{y}} \sim \mathcal{N}(\mu_Y(\tilde{\mathbf{x}}), \sigma_Y^2(\tilde{\mathbf{x}})\mathbf{I}) \quad (11)$$

$$\mu_Y(\mathbf{x}) = \mathbf{Y}^T \mathbf{K}_Y^{-1} \mathbf{k}_Y(\mathbf{x}) \quad (12)$$

$$\sigma_Y^2(\mathbf{x}) = \mathbf{k}_Y(\mathbf{x},\mathbf{x}) - \mathbf{k}_Y(\mathbf{x})^T \mathbf{K}_Y^{-1} \mathbf{k}_Y(\mathbf{x}) \quad (13)$$

where vector $\mathbf{k}_Y(\mathbf{x})$ contains $k_Y(\mathbf{x}, \mathbf{x}_i)$ in its i-th entry, and $\mathbf{x}_i$ is the i-th training vector.

After predicting all body parts $\{\tilde{\mathcal{P}}\}_{1\leq k\leq P}$ and global rigid displacement parameters (i.e. $\{\tilde{\mathbf{y}}^k\}_{1\leq k\leq P}$) for a given pose (i.e frame), we adopt the strategy proposed in [4] to stitch the body mesh back together automatically. We rigidly align each child part to the root part with respect to the overlap region, then we find the closed curve with the least deformation cost between its two instantiations in the root and child geometries, and use it as an optimal boundary for Poisson mesh merging [16] [39]. This process is reiterated for all overlap regions to recover the fully merged body mesh, which is positioned subsequently according the current global rigid displacement prediction.

7. Shape motion parametrization

Once we have learned the motion latent space, we can synthesise infinite variations of any sequence used to build this space using the method detailed in section 6. In this section, we present a method that allows variation synthesis for any blended version of logically compatible pair of input sequences, without the need for costly non-linear interpolation of many input sequence variations in the mesh domain.

Given a pair $\mathbf{Y}_1$ and $\mathbf{Y}_2$ of input sequences exhibiting variations of a common movement, such as a pair of walking and running sequences which both represent locomotion but with two different speeds, seminal work on mesh animation [9] proposes to synthesise new intermediate sequences of the same movement, ranging from the first motion $\mathbf{Y}_1$ to the second motion $\mathbf{Y}_2$, by blending the input sequence pair with weights w varying between 0 and 1: $\mathbf{Y}_w = (1-w)\mathbf{Y}_1 + w\mathbf{Y}_2$. However, for a given parameter value w, only one unique interpolated sequence $\mathbf{Y}_w$ is obtained. Our goal is to perform a similar motion parametrization scheme, but in addition, we want to be able to generate infinite variations for each blended motion with a given parameter w.

One solution is to generate variations $\{\tilde{\mathbf{Y}}_1^l\}$ and $\{\tilde{\mathbf{Y}}_2^m\}$ of input sequence pair $\mathbf{Y}_1$ and $\mathbf{Y}_2$, and each element in the set of blended variations in the observation space $\{(1-w)\tilde{\mathbf{Y}}_1^l + w\tilde{\mathbf{Y}}_2^m\}_{l,m}$ could be considered as a variant of $\mathbf{Y}_w$. However this requires performing mesh non linear interpolation of a big number of sequence pairs for visually plausible outputs, since linear interpolation results in mesh distortions. The work of [8] elaborates on the time cost of non-linear mesh interpolation, which can be achieved using standard algorithms such as [38]. For interpolating meshes from DAN [9] dataset for instance, these approaches cost 3 orders of magnitude more than linear interpolation, which makes them in turn unfit for real time synthesis.

As an alternative solution, we could consider interpolation in the latent space. That is, we could attempt to generate variations of the blended trajectory $\mathbf{X}_w = (1-w)\mathbf{X}_1 + w\mathbf{X}_2$ in the latent space. Conversely, we could generate variations $\{\tilde{\mathbf{X}}_1^l\}$ and $\{\tilde{\mathbf{X}}_2^m\}$ of the latent trajectories $\mathbf{X}_1$ and $\mathbf{X}_2$, and each element in the set of blended variations in the

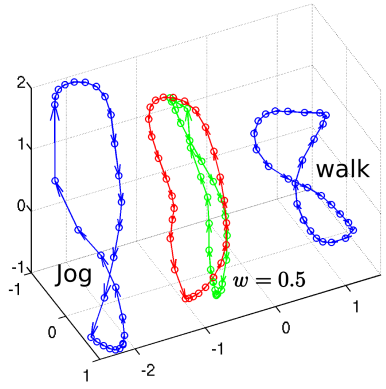


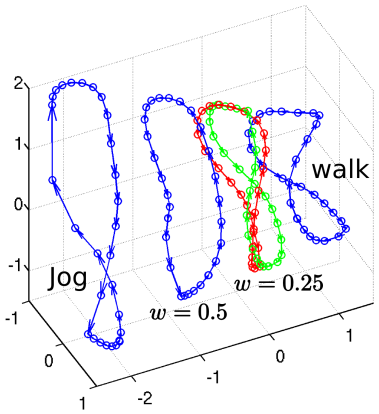
Figure 6: In green: Interpolation of *Walk* (blue) and *Jog* (blue) in latent space. In red: Latent trajectory of non-linearly blended *Walk* and *Jog* in mesh domain.

latent space $\{(1-w)\tilde{\mathbf{X}}_1^l + w\tilde{\mathbf{X}}_2^m\}_{l,m}$ could be considered as a variant of $\mathbf{X}_w$. However, these two strategies suffer from the two following shortcomings:

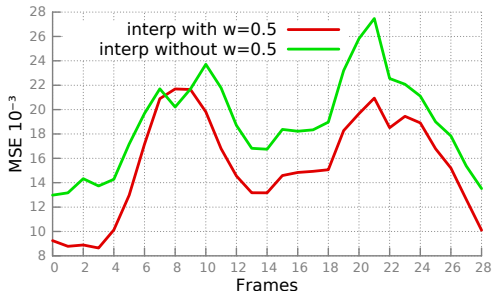
1. Blended motions in the observation space are not necessarily represented by interpolated trajectories with the same proportions in the latent space. For instance in figure 6, we build a GPDM with 5 sequences: *Walk*, *Jog*, *Left*, *Right*, and a blended version of sequences *Walk* and *Jog* with $w = 0.5$. We notice that this sequence's latent trajectory (in red) does not coincide with the linear interpolation of *Walk* and *Jog* trajectories (in green) in the latent space, due to the over representation of walking like poses (*Walk*, *Left* and *Right*) in the training set in this example for instance.

2. In our experiments, sampling variations from a latent trajectory that was not learned with the GPDM model, such as an interpolated trajectory $\mathbf{X}_w$, usually results in degenerate sequences. Hence we limit sampling initialization to trajectories of real sequences that took part of the training process.

In light of these observations, we propose the following solution to overcome the limitations above. We perform blending offline for few discriminative weighting values, e.g. $w \in W = \{0.25, 0.5, 0.75\}$. In our experiments, one intermediate value was enough ($W = \{0.5\}$) to obtain visually compelling results. The GPDM is learned with both the original sequences $\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_m$ and the blended sequences $\{\mathbf{Y}_w\}_{w \in W}$. Variations are generated for latent trajectories of all of these sequences offline. Then, for a new requested blend value $w^* \in [0, 1]$ we can obtain variations of sequence $\mathbf{Y}_{w^*}$ online as follows: First, we find the closest bounding values of w^* in W: $w^* \in [w_1, w_2]$ where $\mathbf{X}_{w=0} := \mathbf{X}_1$ and $\mathbf{X}_{w=1} := \mathbf{X}_2$. Second, we select two variants of the corresponding latent trajectories $\tilde{\mathbf{X}}_{w_1}^l$ and $\tilde{\mathbf{X}}_{w_2}^m$ randomly and blend them accordingly : $\tilde{\mathbf{X}}_{w^*} = \frac{w^*-w_2}{w_1-w_2}\tilde{\mathbf{X}}_{w_1}^l + \frac{w^*-w_1}{w_2-w_1}\tilde{\mathbf{X}}_{w_2}^m$. Finally, we generate an example shape sequence $\tilde{\mathbf{Y}}_{w^*}$.



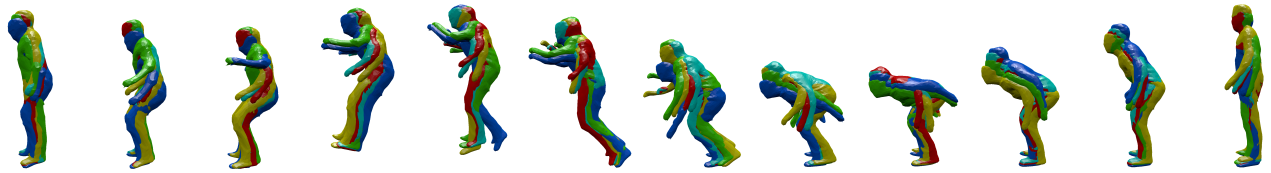
(a) Obtaining latent sequence $w = 0.25$ by: (green) interpolating sequences *Walk* and *Jog*, (red) interpolating nearby sequences *Walk* and $w = 0.5$.



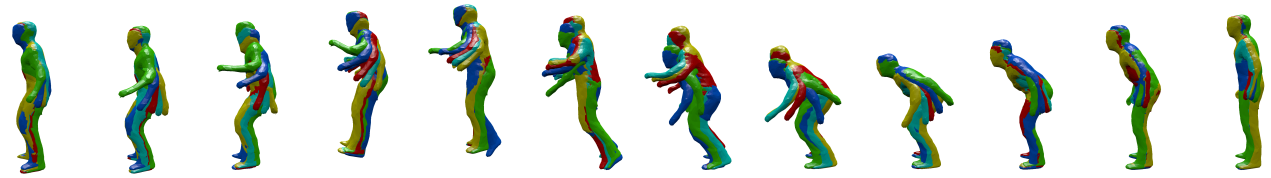
(b) Mean vertex distance between non-linearly interpolated mesh sequence $w = 0.25$ (ground-truth) and: (green) reconstructing interpolation of latent sequences *Walk* and *Jog*, (red) reconstructing interpolation of nearby latent sequences *Walk* and $w = 0.5$.

Figure 7: Interpolation in latent space with and without intermediate learned latent trajectories.

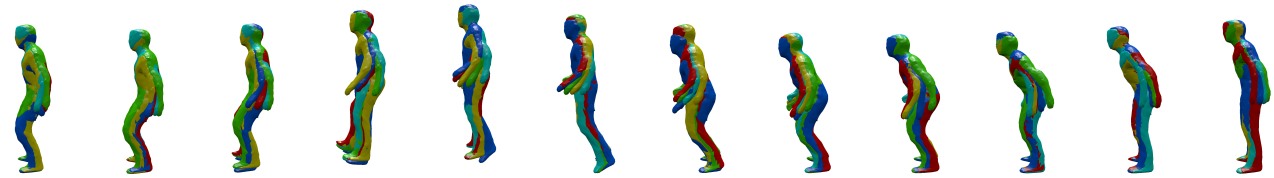
Figure 7 shows that guiding interpolation of latent trajectories with nearby learned intermediate trajectories results in good approximations of non-linearly interpolated mesh sequences. In this particular example, we learn a GPDM with sequences *Walk*, *Jog*, *Left*, *Right* and a blended version of sequences *Walk* and *Jog* with $w = 0.5$. As shown in figure 7a, we generate a sequence $\mathbf{X}_{w=0.25}$ both from interpolating latent trajectories $\mathbf{X}_1$ (*Walk*) and $\mathbf{X}_2$ (*Jog*) that we plot in red, and interpolating nearby trajectories $\mathbf{X}_1$ and $\mathbf{X}_{w=0.5}$ that we plot in green. We realise that the reconstruction of the interpolation that uses intermediate sequence $\mathbf{X}_{w=0.5}$ is closer, in vertex distance, to the non-linearly blended sequence $\mathbf{Y}_{w=0.25}$ in the observation space, that we consider to be ground truth, as shown in figure 7b.



(a) Jumping long.

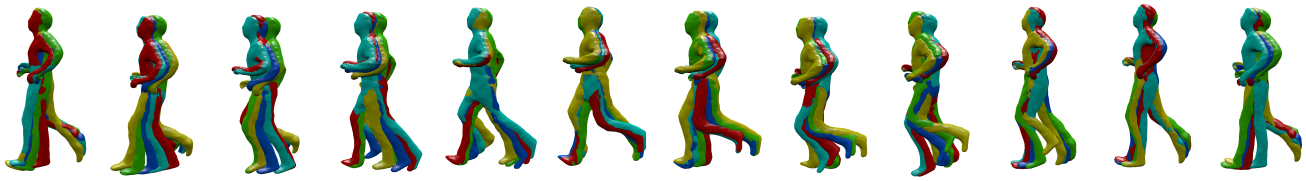


(b) $(1-w)$ Jumping short + w Jumping long, $w = 0.5$.

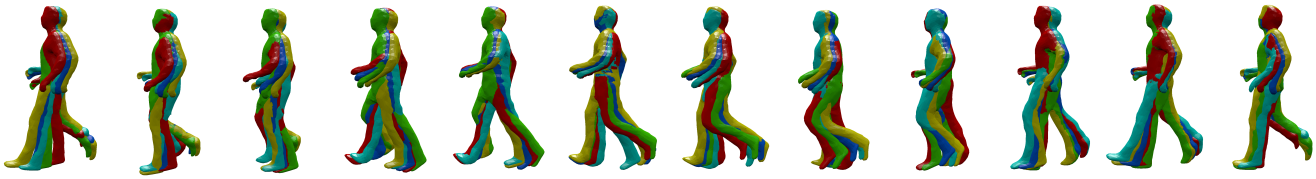


(c) Jumping short.

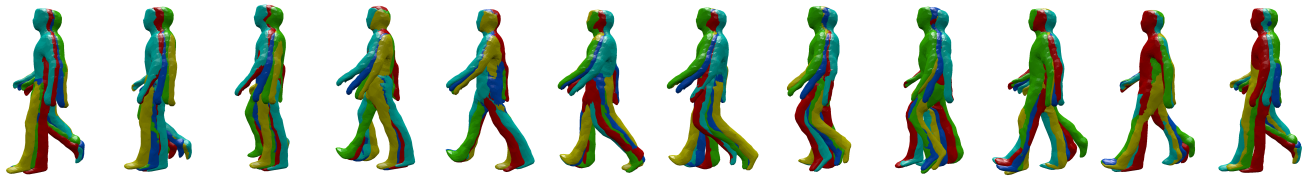
Figure 8: 5 overlapped Jumping variations, each with a different color.



(a) Jogging.



(b) $(1-w)$ Walking + w Jogging, $w = 0.5$.



(c) Walking.

Figure 9: 5 overlapped locomotion variations, each with a different color.

8. Results

We use DAN surface motion capture dataset [9] to evaluate our method. Meshes have 2667 vertices and 5330 faces and motions are recorded at 25 fps. We build two GPDMs using locomotion sequences and jumping sequences respectively.

Figure 8 shows results from the GPDM of jumping movements, which comprises sequences *Short jump*, *Long jump*, *Low jump*, *High jump*. All sequences are temporally aligned to *Short jump* with dynamic time warping

guided with feet contact annotation. To demonstrate motion parametrization, we interpolate sequences *Short jump* and *Long jump* non-linearly with $w = 0.5$ and add this interpolated mesh sequence to the training sequences. As a result, we can generate variations of sequences *Short jump*, *Long jump*, *Low jump*, *High jump* and blended versions of sequences *Short jump* and *Long jump* with any weighting proportions. We show in this figure variations of training sequences and also blended sequences.

In figure 9, we show results from the GPDM of locomotion movements, which comprises sequences *Walk*, *Jog*, *Left turn*, *Right turn*. All sequences are temporally aligned to *Walk* with dynamic time warping guided with feet contact annotation. To demonstrate motion parametrization, we interpolate sequences *Walk* and *Jog* non-linearly with $w = 0.5$ and add this interpolated mesh sequence to the training sequences. As a result, we can generate variations of sequences *Walk*, *Jog*, *Left turn*, *Right turn* and blended versions of sequences *Walk* and *Jog* with any weighting proportions. We show in this figure variations of training sequences and also blended sequences.

As we can see in figures 8 and 9 and the accompanying video, the output variations are logically similar but slightly different from each other in poses and timings. The differences between the variations are big enough to be noticed by users, but still conserve the main characteristics of the base movement. The realism of the generated sequences matches that of the input ones, as poses and dynamics are overall sound and coherent.

9. Conclusion

We presented in this work a solution for generating infinite variations of a subject movement using few training sequences of surface motion capture, based on Gaussian Process Dynamical Models. We also contributed an algorithm that allows synthesis of variations for any blended version of the input sequences without costly non-linear interpolation of many motion sequence variations in mesh domain. While the differences between a movement variations are easily noticeable, these generated motions are mostly visually plausible and match the realism level of the input sequences. As a next step, this work can be extended to model and synthesis both shape and appearance variation.

References

[1] B. Allain, J.-S. Franco, and E. Boyer. An efficient volumetric framework for shape tracking. In *CVPR*, 2015. 1

[2] B. Bodenheimer, A. V. Shleyfman, and J. K. Hodgins. The effects of noise on the perception of animated human running. In *Computer Animation and Simulation*, volume 99, 1999. 2

[3] A. Boukhayma and E. Boyer. Video based Animation Synthesis with the Essential Graph. In *3DV*, 2015. 1

[4] A. Boukhayma, J.-S. Franco, and E. Boyer. Surface motion capture transfer with gaussian process regression. In *CVPR*, 2017. 1, 3, 5

[5] A. Boukhayma, V. Tsiminaki, J.-S. Franco, and E. Boyer. Eigen Appearance Maps of Dynamic Shapes. In *ECCV*, 2016. 1

[6] C. Budd, P. Huang, M. Klaudiny, and A. Hilton. Global non-rigid alignment of surface sequences. *IJCV*, 102(1-3), 2013. 1

[7] D. Casas, M. Tejera, J.-Y. Guillemaut, and A. Hilton. Parametric control of captured mesh sequences for real-time animation. In *MIG*, 2011. 2

[8] D. Casas, M. Tejera, J.-Y. Guillemaut, and A. Hilton. Interactive animation of 4d performance capture. *TVCG*, 19(5), 2013. 5

[9] D. Casas, M. Volino, J. Collomosse, and A. Hilton. 4D Video Textures for Interactive Character Appearance. *CGF (Proceedings of EUROGRAPHICS)*, 33(2), 2014. 1, 2, 4, 5, 7

[10] A. Collet, M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, and S. Sullivan. High-quality streamable free-viewpoint video. *ACM TOG*, 34(4), 2015. 1

[11] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, P. Kohli, V. Tankovich, and S. Izadi. Fusion4d: Real-time performance capture of challenging scenes. *ACM TOG*, 35(4), 2016. 1

[12] M. Gleicher, H. J. Shin, L. Kovar, and A. Jepsen. Snap-together motion: assembling run-time animations. In *ACM SIGGRAPH classes*, 2008. 2

[13] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popović. Style-based inverse kinematics. *ACM TOG*, 23(3), 2004. 2

[14] C. M. Harris and D. M. Wolpert. Signal-dependent noise determines motor planning. *Nature*, 394(6695), 1998. 2

[15] R. Heck and M. Gleicher. Parametric motion graphs. In *ACM I3D*, 2007. 2

[16] X. Huang, H. Fu, O. K.-C. Au, and C.-L. Tai. Optimal boundaries for poisson mesh merging. In *SPM*, 2007. 5

[17] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. In *ECCV*, 2016. 1

[18] L. Kovar and M. Gleicher. Automated extraction and parameterization of motions in large data sets. *ACM TOG*, 23(3), 2004. 2

[19] M. Lau, Z. Bar-Joseph, and J. Kuffner. Modeling spatial and temporal variation in motion data. *ACM TOG*, 28(5), 2009. 1, 2

[20] N. D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *NIPS*, 2004. 2

[21] W. Ma, S. Xia, J. K. Hodgins, X. Yang, C. Li, and Z. Wang. Modeling style and variation in human motion. In *ACM SCA*, 2010. 1, 2

[22] D. J. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003. 3, 4, 5

[23] M. Müller. *Information Retrieval for Music and Motion*. Springer-Verlag New York, Inc., 2007. 4

[24] A. Mustafa, H. Kim, J.-Y. Guillemaut, and A. Hilton. Temporally coherent 4d reconstruction of complex dynamic scenes. In *CVPR*, 2016. 1

[25] R. M. Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012. 3

[26] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *CVPR*, 2015. 1

[27] K. Perlin. Real time responsive animation with personality. *TVCG*, 1(1), 1995. 2

[28] F. Prada, M. Kazhdan, M. Chuang, A. Collet, and H. Hoppe. Motion graphs for unstructured textured meshes. *ACM TOG*, 35(4), 2016. 1

[29] K. Pullen and C. Bregler. Animating by multi-level sampling. In *Computer Animation*, 2000. 2

[30] K. Pullen and C. Bregler. Motion capture assisted animation: Texturing and synthesis. *ACM TOG*, 21(3), 2002. 2

[31] J. Starck and A. Hilton. Surface capture for performance-based animation. *IEEE CGA*, 27(3), 2007. 1

[32] C. Stoll, J. Gall, E. De Aguiar, S. Thrun, and C. Theobalt. Video-based reconstruction of animatable human characters. *ACM TOG*, 29(6), 2010. 1

[33] D. Vlasic, P. Peers, I. Baran, P. Debevec, J. Popović, S. Rusinkiewicz, and W. Matusik. Dynamic shape capture using multi-view photometric stereo. *ACM TOG*, 28(5), 2009. 1

[34] M. Volino, D. Casas, J. Collomosse, and A. Hilton. Optimal representation of multiple view video. In *BMVC*, 2014. 1

[35] J. Wang, A. Hertzmann, and D. M. Blei. Gaussian process dynamical models. In *NIPS*, 2006. 2, 4

[36] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *PAMI*, 30(2), 2008. 1, 2, 3, 4

[37] R. Wang, L. Wei, E. Vouga, Q. Huang, D. Ceylan, G. Medioni, and H. Li. Capturing dynamic textured surfaces of moving targets. *ECCV*, 2016. 1

[38] D. Xu, H. Zhang, Q. Wang, and H. Bao. Poisson shape interpolation. *Graphical models*, 68(3), 2006. 5

[39] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, and H.-Y. Shum. Mesh editing with poisson-based gradient field manipulation. *ACM TOG*, 23(3), 2004. 5