Surface Motion Capture Transfer with Gaussian Process Regression

Adnane Boukhayma, Jean-Sébastien Franco, Edmond Boyer

Surface Motion Capture Transfer with Gaussian Process Regression

Adnane Boukhayma, Jean-Sébastien Franco, Edmond Boyer
Inria, LJK, Univ. Grenoble Alpes
firstname.lastname@inria.fr

Abstract

We address the problem of transferring motion between captured 4D models. We particularly focus on human subjects for which the ability to automatically augment 4D datasets, by propagating movements between subjects, is of interest in a great deal of recent vision applications that builds on human visual corpus. Given 4D training sets for two subjects for which a sparse set of corresponding key-poses are known, our method is able to transfer a newly captured motion from one subject to the other. With the aim to generalize transfers to input motions possibly very diverse with respect to the training sets, the method contributes with a new transfer model based on non-linear pose interpolation. Building on Gaussian process regression, this model intends to capture and preserve individual motion properties, and thereby realism, by accounting for pose inter-dependencies during motion transfers. Our experiments show visually qualitative, and quantitative, improvements over existing pose-mapping methods and confirm the generalization capabilities of our method compared to state of the art.

1. Introduction

After more than a decade of predominance of standard motion capture, we have witnessed in the recent years the emergence of 4D surface capture systems either through high quality multi-view set-ups [9, 10] or in a more democratized form with low-cost sensors [16, 13, 27]. The tasks related to the acquisition and exploitation of 4D data, such as tracking [1, 15, 7], appearance modelling [6, 24] and animation [17, 8, 5] have received a great deal of interest from the vision and graphics communities recently, and there are still many problems to be solved in this respect. Notably, a number of challenges remain open when dealing with corpus of moving subjects. Such datasets can find many uses for 3D content generation, augmenting machine learning training sets or statistical modeling, among other examples.

In this paper, we address the task of transferring captured motions from one subject to another, to the benefit of enabling the generation of uncaptured animations, removing the burden of exhaustive motion acquisition for each subject, and broadening creative possibilities with captured 4D models. Given two training sets of motions for two subjects and a sparse set of annotated semantic correspondences between the two sets, a regression model can be built, by which a new captured motion associated to one of the subjects can be transferred to the other. While the transfer for sparse surface parameterizations such as motion capture data have received some attention [19, 23, 29, 11], only a handful of works exists that deal with the particular problem of surface to surface transfer relevant to our shape capture situation [20, 3]. The most successful approach to date [3] obtains inspiring results with a linear representation of pose mapping.

We propose a more general model, which significantly broadens transferring capabilities in a multi-view capture context, thanks to the following two important contributions. First, we propose a novel motion mapping model, based on non-linear Gaussian Process regression and on body part segmentation, which significantly improves the accuracy and generalization abilities of mappings for a given training set. Unlike existing works, this regression model also captures the global rigid component of motion, allowing more realistic transfer of subject displacements. Second, while previous works only use a sparse set of matching key-frames in the training set, we provide a full temporal matching densification approach based on probabilistic dynamic time-warping. Starting from the initially provided matching pairs, or key-frames, between two paired training sequences, it extrapolates correspondences densely between the two sequences. This allows the regression to benefit from an expanded and complete set of frame pairs, which in turn increases realism by better preserving learned individual dynamic patterns in new motions.

In order to conduct evaluation, we consider two annotated datasets acquired with different platforms. These datasets contain motions for two subjects, any subset of which can be used for training and the other for testing. Testing data annotations are used as baseline to measure

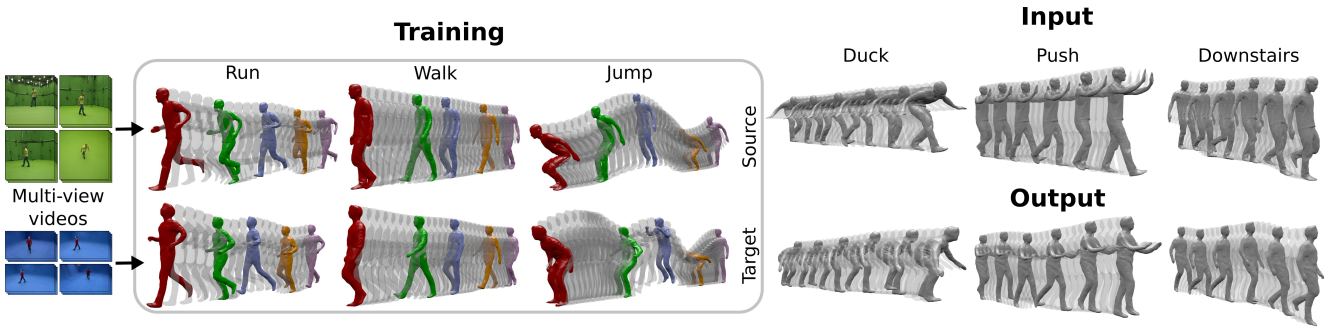


Figure 1: Approach architecture. Left: Multi-View acquisition of motion sequences for 2 subjects; Middle: Training of the motion mapping between subjects given shape sequences with sparse frame correspondences (in color); Right: Motion animations of subject 2 (bottom) given new input motions for subject 1 (top).

the accuracy of the transfer and generalization capability of the method. We also evaluate the benefit of densification versus only using the initially annotated sparse key-frame pairs. Under this protocol, we are able to verify quantitatively and qualitatively that our method outperforms all existing strategies and yields more visually pleasing results in typical vision-based surface capture scenarios, in particular achieving higher fine-scale motion fidelity of the transfer.

2. Related work

Existing works in the field of captured surface motion transfer can be roughly categorized with respect to the surface motion parameterization they consider and to the motion transfer model they apply.

Skeleton to Surface. Articulated motions can be naturally parameterized using skeletons and several works resort to such models to transfer motions between subjects as in [11, 23, 29]. In order to be applicable to surface based information, e.g. 3D meshes in our case, this strategy requires mesh rigging and skinning or tracked control points. Obtaining this information reliably in generic situations is often difficult, for instance with loose clothing, and we follow a different strategy that does not rely on intermediate representations.

Surface to Surface given correspondences. The seminal work of [20] proposed an elegant solution to transfer deformations between triangulated meshes based on deformation gradients. Such a solution was applied to synthetic and real models of humans, animals as well as recently to faces [22]. [4] generalized such deformation transfer to multi-component and non manifold meshes using harmonic maps. While successful, this strategy however requires correspondences between surface meshes which can easily be involved and complicated. In addition, a local transfer strategy can be disadvantageous with shapes that differ in the way they deform locally when undergoing similar global motions. We therefore favor a more global approach.

Surface to Surface without correspondences. The work of [19] uses a linear pose mapping from sparse point clouds to surface data for character control. Closer to our concern, [3] proposed a method that performs semantic deformation transfer between surfaces. This method does not require local correspondences and accomplishes instead motion transfers through poses within spaces spanned by key-poses (key-frames), for which global correspondences are given. This approach was later extended to multi-component objects [31]. We follow a similar strategy, albeit significantly improving the pose space representation, which accounts for all training poses in our case, instead of sparse key-poses only. Ours differs also in the transfer function that accounts for global rigid displacements and is based on non-linear regressions instead of direct pose mapping, with the benefit of allowing the transfer of more complex motion patterns.

Transfer Model. Motion transfer models have been largely studied in the literature in the case of Mocap and therefore sparse data. They range from direct transfer to linear and non-linear interpolation models. Among the latter, Gaussian Processes have proven to be effective for performing various tasks with Mocap data, such as non-trivial transfers to non-humanoid characters [29] or, interestingly with complex real subjects, motion style and variability modelling [25, 26]. Our framework thus builds on Gaussian Process Regression and extends it to surface mesh data.

3. Method Overview

Our approach considers as input 3D shape sequences of two moving subjects as acquired with multi-view acquisition systems (see figure 1). It handles the possibility of different acquisition systems for source and target, *e.g.* different resolution and frame rate. Shapes are represented by 3D

meshes which are globally consistent for a given subject, *i.e.* all the subject's poses are represented with the same mesh. This is not true between subjects that can even present different topologies. A first set of various motions is used to train the transfer function between subjects. We assume that sparse frame correspondences between sequences (as illustrated with colors in figure 1) are given to bootstrap the motion transfer. They represent semantic correspondences [3] between subject's poses as seen or desired by the user and therefore need not be accurate. The transfer mapping estimation follows then two steps:

1. Correspondence Densification: from the sparse semantic correspondences, or key-poses, a dense correspondence map between the two subject's poses is obtained using dynamic time warping applied to motion sequences in the training set.

2. Transfer Model: a non-linear mapping between the subjects' pose spaces is learned using Gaussian Process Regression applied on the full set of pose correspondences.

Given then a newly acquired motion sequence for one of the subjects, its *transferred* version onto the other subject is obtained by mapping each pose of the source sequence to a corresponding pose in the space of the target subject using the transfer model. In practice, shapes are represented as body parts over which regressions are performed. The benefit is to significantly improve the transfer model accuracy since the limbs of observed subjects can move independently, making global regression over the full body less precise. Another feature of the approach is to compensate the increased complexity with body part regressions using PCA dimension reduction applied to body parts. We introduce the different components of the approach below.

4. Shape Pose Representation

A shape is represented by a 3D mesh $\mathcal{M}$, whose pose is encoded in its vertex positions. Different subjects are represented by different meshes. A shape pose is in practice characterized by elements such as its global rigid transformations and its body part poses. While not fully independent, only weak dependencies of body limb movement with body motions are observed and the body-part model produces better results in our experiments. Consequently, we choose to learn transfer functions for all elements using independent regressions. In addition, we further reduce body part pose representations using PCA.

Rigid Transformation Let $\{\mathcal{M}_i\}_{1\leq i\leq n}$ be the n poses of a shape $\mathcal{M}$. We first rigidly align all the consistent meshes $\{\mathcal{M}_i\}_{1\leq i\leq n}$ of a subject using standard Procrustes analysis applied to the mesh vertex coordinates. We assume, without loss of generality, that all motions starts at the same space location and with the same initial orientation. Hence, a shape pose is represented by its aligned mesh $\bar{\mathcal{M}}_i$ and its globally rigid displacement from the previous pose in the motion sequence in which it appears. This elementary rigid motion is composed of a translation δT_i and rotation δR_i that will further be part of the transfer analysis. To this aim, in particular to perform regression, a 6-dimensional linear parametrization of the transformation $(\delta T_i, \delta R_i) \mapsto (t_x, t_y, t_z, h_1, h_2, h_3)^T$, based on exponential maps for the rotation parameters h_i, is used.

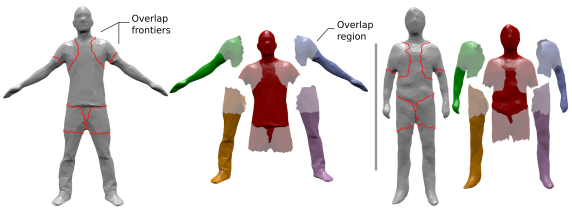


Figure 2: Body parts: shaded colors represent overlapping regions between parts delimited by user given curves.

Body Parts As mentioned earlier, body parts can move independently, for instance arms can move differently over various instances of a walking movement while legs present similar motions. As a result, learning independent transfer functions for each body part can increase accuracy compared to a global strategy, which is confirmed by our experiments (see section 7). In order to ease the decomposition in practice while keeping robustness, we adopt a strategy similar to [21] with a coarse but overlapping mesh segmentation. To this purpose, as illustrated in figure 2, for each subject the user provides closed and non-intersecting curves that delimit each overlapping region between contiguous body parts on $\mathcal{M}$. Throughout the regression process, each body part will be augmented with the overlap regions it shares with its neighboring parts. During the motion transfer, merging will be achieved to ensure seamless body part stitching as described later in section 6.1.

Each aligned mesh $\bar{\mathcal{M}}_i$ is therefore decomposed into N body part sub-meshes $\{\mathcal{P}_i^k\}_{1\leq k\leq N}$. In our experiments, we use $N = 5$ parts in a tree structured hierarchy including a torso as the root, and a pair of arms and legs as children nodes.

PCA Dimension Reduction In order to reduce the computational cost, dimension reduction is applied on each body part using Principal Component Analysis. All body part sub-meshes for a given subject are first rigidly aligned and PCA applied on the vertex locations. For a given pose i, each body part $\mathcal{P}_i^k$ is now on expressed as a vector of eigen decomposition coefficients $\mathrm{x} = (x_1, \ldots, x_m)^T$, where the reduced dimension m could be different from source to target subjects. In practice, we use $m = 20$ coefficients for each body part. Note that this operation, as the body segmentation step, introduces additional inaccura-

cies in the overlapping regions that require post-processing (section 6.1).

5. Pose Mapping

Given two groups of shape poses for two subjects and a set of key-pose correspondences between the groups, we want to estimate a function that maps poses from one group to the other. Since the objective is to transfer motions, hence sequences of poses, we expect this function to provide consistent and realistic temporal pose arrangements. Relying for that purpose on sparse key-poses only, as in [3], is sub-optimal when global semantic motion correspondences, e.g. walking or running, are known. We therefore first extend the set of key-pose correspondences to full pose correspondences between similar motions. Second, in order to better capture pose inter-dependencies, we perform a global non-linear regression over all pose correspondences. Our experiments demonstrate that both contribute to a better accuracy of motion transfers.

5.1. Temporal Correspondence Densification

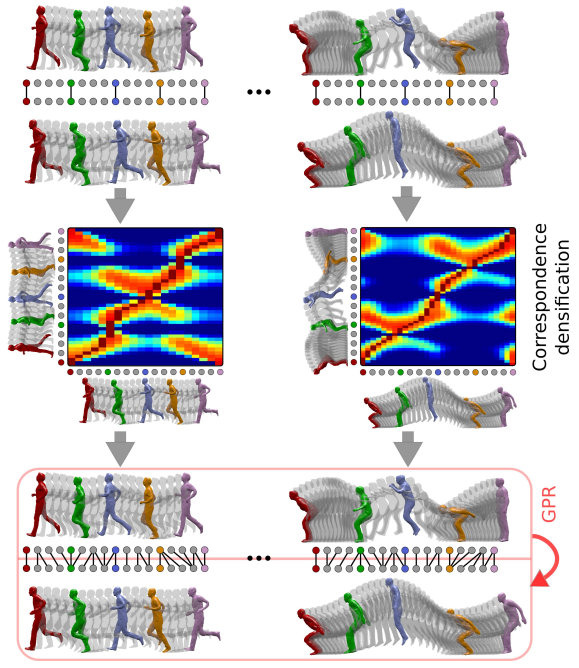


Figure 3: Pose correspondence densification and mapping.

For every pair of matching motions in the training set, sparse semantic correspondences between the source and target poses are provided by the user (see figure 3). Starting from these initial correspondences, our aim is to optimally propagate associations to the rest of the sequence by solving a shortest path problem using dynamic time warping. The costs of source and target pose associations are derived from a linear mapping between poses as learned from the initial correspondences. A linear model per motion is sufficient here to capture the information given by the very few initial associations (around 5 key-poses per sequence in our experiments).

Association cost Using the notations introduced in section 4, we hence assume that the mapping between the target and source pose vectors y and x of part k is linear, up to an observation noise vector ε, thus:

$$\mathrm{y} = M\mathrm{x} + \varepsilon, \tag{1}$$

where M is the matrix of the linear transformation for part k. We assume that the additive noise follows a Gaussian distribution: $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$, where $\sigma^2 = 0.01$ in our experiments. Together with the above linear formulation, this assumption gives rise to the following likelihood for body part k:

$$p(Y|X, M) = \mathcal{N}(MX, \sigma^2 I), \tag{2}$$

where matrices X and Y contain the stacked source and target initially matched key-poses: $X = (\mathrm{x}_{1^s}, \ldots, \mathrm{x}_{L^s})$ and $Y = (\mathrm{y}_{1^t}, \ldots, \mathrm{y}_{L^t})$ for part k, assuming we have L key-pose correspondences. Given a zero mean identical Gaussian prior on rows of M, the posterior distribution over M can then be written as follows:

$$\begin{aligned} p(M|Y, X) &\propto p(Y|X, M)p(M), \\ p(M|Y, X) &= \mathcal{N}(\tfrac{1}{\sigma^2} Y X^T A^{-T}, A^{-1}), \end{aligned} \tag{3}$$

where $A = \frac{1}{\sigma^2} X X^T + I$. The predictive distribution for the function $M\mathrm{x}$ at x is obtained by averaging over all matrices M with the Gaussian posterior in (3):

$$\begin{aligned} p(M\mathrm{x}|\mathrm{x}, X, Y) &= \int_M p(M|X, Y)\, M\mathrm{x} \\ &= \mathcal{N}(\tfrac{1}{\sigma^2} Y X^T A^{-T}\mathrm{x}, \mathrm{x}^T A^{-1}\mathrm{x} I). \end{aligned} \tag{4}$$

Hence for a given body part k, the probability of the target pose y_j to belong to the predictive distribution for $M\mathrm{x}$ at the source pose x_i is then:

$$P^k_{i,j} = e^{-\frac{1}{2}(\mathrm{y}_j - \mu_i)^T (\Sigma_i)^{-1} (\mathrm{y}_j - \mu_i)}, \tag{5}$$

where $\mu_i = \frac{1}{\sigma^2} Y X^T A^{-T}\mathrm{x}_i$ and $\Sigma_i = \mathrm{x}_i^T A^{-1}\mathrm{x}_i I$. For the whole body, the probability, or cost, of this association between poses i and j is then the product of the latter probabilities over all the body parts:

$$P_{i,j} = \prod_{1 \le k \le N} P^k_{i,j}. \tag{6}$$

Dense Correspondence Denoting by f_s and f_t the source and target sequence sizes, the next step is to find a map $\phi : [\![1, ..., f_s]\!] \rightarrow [\![1, ..., f_t]\!]$ between source and target sequence poses that maximizes the product of the association probabilities $\{(i, \phi(i))\}$:

$$\begin{aligned} \phi_* &= \underset{\phi}{\text{argmax}} \prod_i P_{i,\phi(i)}, \\ &= \underset{\phi}{\text{argmin}} \sum_i -\log\left(P_{i,\phi(i)}\right). \end{aligned} \tag{7}$$

When formulated as a minimization, as above, and taking into account some motion priors such as continuity and monotonicity, finding a mapping ϕ can be cast as a shortest path problem, solved using a dynamic programming algorithm [14]. The solution corresponds to a least cost path within a cost matrix (shown in figure 3) taking the negative log of the probability $P_{i,j}$ as value for node (i, j). In our implementation, we bound the path through the cost matrix to be continuous, not reserve path, not include more than 3 vertical or horizontal consecutive nodes in a row, and also run through the initial key-pose correspondences.

5.2. Pose and Displacement Mapping

Using the densification approach presented before, we can augment the set of source and target pose correspondences using all the motion sequences present in the training datasets. This allows to benefit from both variability and redundancy in the shape pose training set. In the previous section, a linear model was used to map poses between the source and target in each motion sequence. We consider now the full set of corresponding poses, whose correspondence distribution is expected to be better captured by a more elaborate model, as demonstrated in our experiments.

Non-linear Model For a given body part, we assume that the i-th parameter y_i of the target pose y can be related to a new source vector x with a non-linear function and up to an observation noise:

$$y_i = f_i(\mathrm{x}) + \varepsilon, \tag{8}$$

with $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$ and $\sigma_n^2 = 0.1$ in our experiments. We use a Gaussian Process (GP) to describe the distribution over such functions $f_i(\mathrm{x})$ given a training set.

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution. It extends multivariate Gaussian distributions to infinite dimensionality. GP have shown to be efficient in solving similar regression problems on sparse motion representations. A GP is defined by its mean and covariance functions. Since body parts are first aligned, we assume the mean functions to be null. For the covariance functions, we introduce the following entities:

$$\begin{aligned} K &= \begin{bmatrix} k(\mathrm{x}_1, \mathrm{x}_1) & \dots & k(\mathrm{x}_1, \mathrm{x}_F) \\ \vdots & \ddots & \vdots \\ k(\mathrm{x}_F, \mathrm{x}_1) & \dots & k(\mathrm{x}_n, \mathrm{x}_F) \end{bmatrix}, \\ K_* &= \begin{bmatrix} k(\mathrm{x}_*, \mathrm{x}_1) & \dots & k(\mathrm{x}_*, \mathrm{x}_F) \end{bmatrix}, K_{**} = k(\mathrm{x}_*, \mathrm{x}_*), \end{aligned} \tag{9}$$

where F is the total number of training poses, and x_* is a new input pose for a body part and $k(.,.)$ a kernel function. In our experiments, the neural network covariance below outperforms other traditional kernels (see section 7):

$$k_{\text{nn}}(\mathrm{x}, \mathrm{x}') = \sigma_f^2 \sin^{-1}\left(\frac{\frac{1}{2l^2}\mathrm{x}^T\mathrm{x}'}{\sqrt{(1+\frac{1}{2l^2}\mathrm{x}^T\mathrm{x})(1+\frac{1}{2l^2}\mathrm{x}'^T\mathrm{x}')}}\right) \tag{10}$$

where the hyper-parameters l and σ_f are optimized using conjugate gradients [18]. Using the GP prior on function f_i, the joint distribution of the observed target pose parameters $\{y_i\}$ over the training set and the predicted parameter $y_{i*} = f_i(\mathrm{x}_*) + \varepsilon$ at the new source pose x_* can be expressed as follows:

$$\begin{bmatrix} y_i \\ y_{i*} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K + \sigma_n^2 I & K_*^T \\ K_* & K_{**} \end{bmatrix}\right). \tag{11}$$

By conditioning the joint Gaussian prior distribution on the observations [18], we obtain the posterior distribution for y_{i*}:

$$\begin{aligned} p(y_{i*}|X, Y_i, \mathrm{x}_*) \sim \mathcal{N}\big(& K_*[K + \sigma_n^2 I]^{-1} Y_i, \\ & K_{**} - K_*[K + \sigma_n^2 I]^{-1} K_*^T \big), \end{aligned} \tag{12}$$

where the matrix X contains the stacked values of the source part poses x in the training set and the vector Y_i contains the stacked values of the pose parameter y_i in the corresponding target pose in the training set. Finally, for a given input pose x_*, we take the mean of this distribution as the predicted output value;

$$y_{i*} = K_*[K + \sigma_n^2 I]^{-1} Y_i, \tag{13}$$

this for all the body part pose parameters $\{y_{i*}\}_{1 \leq i \leq m}$ and over all body parts.

Rigid Displacement As explained in section 4, a shape pose is characterized by its body part poses as well as its rigid displacement from the previous pose in the considered motion sequence. This rigid displacement is modeled using a similar GPR approach that learns from all training correspondences the mappings between the source displacement vectors and each of the 6 target displacement vector values.

6. Shape Pose Reconstruction

Using the approach described in the previous sections we can predict body part poses and global rigid displacement for any input source pose. Since these predictions are

independent, a post-processing step is required to combine them all in a consistent way. This includes body part correction, body part stitching and a global rigid displacement according to the prediction. We elaborate on the body part tasks below. Note also that final motions might require additional traditional post-processing such as self-intersection clean up, or foot-skating correction to ensure foot contacts don't appear to be slipping.

6.1. Body Part Correction

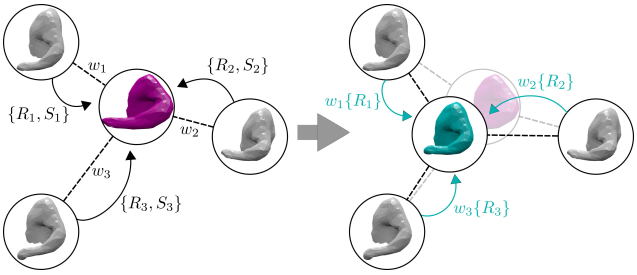


Figure 4: Body part correction algorithm.

As a result of the reduced pose parametrization, mesh deformities can appear in the predicted body-part meshes such as shrinkage, distortion and local magnification. We propose in this section an example-based algorithm that solves these limitations. It relies on the assumption that mesh deformities result from strong non-isometric deformations, that we therefore try to factor out. The algorithm is composed of the following steps, illustrated is figure 4:

- The M nearest meshes to the predicted mesh are found in the training set, according to a simple vertex-to-vertex distance. In our experiments, 3 nearest meshes were enough to obtain perceptually valid results. A weight w_i is associated to each nearest mesh, which decreases with the distance to the predicted mesh.

- Polar decompositions are performed between triangles on nearest meshes and their correspondents on the predicted mesh to obtain isometric $\{R_i\}$ (Rotation) and non-isometric $\{S_i\}$ (Shear/Scale) deformation components per triangle.

- Nearest mesh triangles are deformed with only the isometric per-face components $\{R_i\}$ and the corresponding deformed meshes are reconstructed using a Poisson based method [28]. The resulting M deformed meshes are then combined using weights $\{w_i\}$.

Figure 5 shows a correction example where the algorithm helps correct shrinkage and local magnification in the prediction.

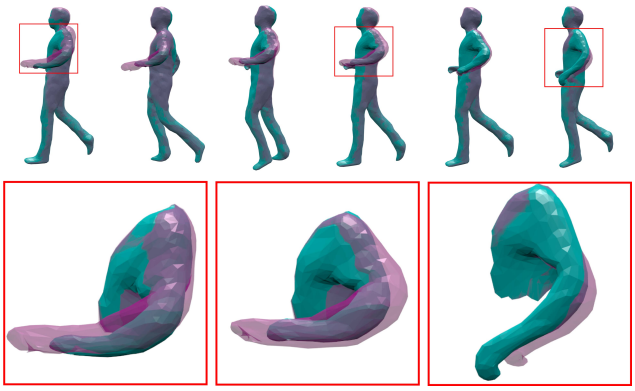


Figure 5: Example of body part correction (in green).

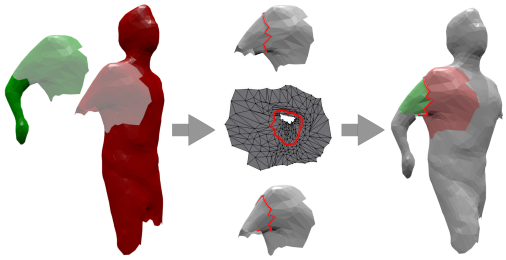


Figure 6: Body part stitching.

6.2. Part stitching

Assuming we deal with zero-genus manifold meshes, drawing two non-intersecting curves on the mesh will result in an overlap region that is topologically equivalent to a cylinder, as can be visualized in figure 6 flattened with conformal mapping. For a pair on contiguous body parts, and after rigidly aligning the child parts to the root part with respect to the overlap region, we look for a closed curve within this shared overlap topology that we will use as a Dirichlet boundary condition for Poisson mesh merging algorithm [30] applied to these two parts. The topological curve that is more likely to produce seamless merges is the one with the least deformation cost between its two instantiations in the root and child geometries. Following [12], an approximation of this solution is found with a Dijkstra shortest path algorithm. This process is subsequently reiterated for all overlap regions to recover the full final body merged mesh.

7. Evaluation

To demonstrate our method, we use two datasets containing temporally coherent mesh sequences of matching basic motions, and we attempt motion transfer from one dataset to the other in both ways. DAN [8] dataset has meshes with 2667 vertices and 5330 faces and was recorded at 25 fps. THOMAS [5] dataset has more uniform meshes with 5000 vertices and 10000 faces and was recorded at a higher

frame rate; 50 fps. For the training, we match THOMAS's sequences *Walk*, *Run*, *Jump*, *Jump forward* and *Bend* with DAN's sequences *Walk*, *Jog*, *High jump*, *Long jump* and *Big box low* respectively. We initialize the key-pose correspondences with 5 frames for each sequence and end up with a total number of 350 pairs of frame correspondences after densification. As we show in figures 10,1 and the accompanying video, we succeed in transferring both motions that are semantically similar to the ones in the training set, like *Run2walk*, and other new motions not represented in the training set, such as *Duck*, *Push*, *Upstairs*, *Downstairs*.

For quantitative evaluations, we fully annotate the motion datasets with frame to frame correspondences, and we randomly split this ground-truth data into a training set and a cross-validation set. The training set accounts for 70% of the pair samples, while the remainder belongs to the testing set. For a given pose regression method, we plot the training set error and the cross-validation set error as a function of the size of the training subset and in terms of root mean square of mesh per-vertex error. We carry these experiments for transfer in both direction, i.e. from THOMAS to DAN and from DAN to THOMAS.

Transfer Model We compare our pose regression model to other methods both quantitatively and qualitatively. We use Gaussian Process regression with the neural network kernel, and compare it to the squared exponential kernel, the linear kernel and the method presented in [3], which uses source and target affine spaces spanned with key-poses, and a mapping derived from these key-pose pair associations. As shown in figure 7, our method outperforms the rest for both training and testing errors.

The low cross-validation error curve of our method is demonstrated in the qualitative comparison through a visually superior generalization ability that we can witness in figure 10. For both examples, *Duck* and *Run2walk*, our method succeeds in extrapolating the best prediction corresponding to the input motion, while preserving the properties inherent to the target motion space. The other methods fail to achieve comparable results. We note that we show raw outputs in figure 10 for all methods alike, prior to the mesh enhancement process described in section 6.1 (see also accompanying video).

Perceptually, we noticed that affine mapping performs particularly poorly in our training scenario, which consists of relatively large training examples obtained from dense temporal matching between several different motions. On the contrary, our strategy does not suffer from the training set extension thanks to the Gaussian model ability to better handle redundancy, variability and uncertainty in the associations.

Dense correspondence We evaluate the impact of using dense correspondence instead of sparse key-pose associations both qualitatively and quantitatively. To this end, we recalculate the learning curves in figure 7. Only this time, the training subset, randomly selected with gradually increased size throughout the experiment, is initialized with key-pose pairs. We can see in figure 8 that the generalization error globally decreases for all methods as we densify the correspondences, which accordingly translates into visually improved predictions in figure 10. The linear regression seems to be an exception to this behaviour. In fact, with more training examples, the linear model under-fits the learning set, which also results in poor predictions. We note also that cross-validation errors are globally lower in figure 8 which makes perfect sense as key-poses are meant to better encode variability in the data and hence yield better generalization.

Body parts Using our regression method, we reiterate the learning curve experiments (figure 7) for two different pose representations: Using the whole body mesh, and using body parts. As we show in figure 9, independent transfer of body part motion compares favorably to the whole body approach both in training and testing errors, for equivalent total number of input and output dimensions.

8. Conclusion

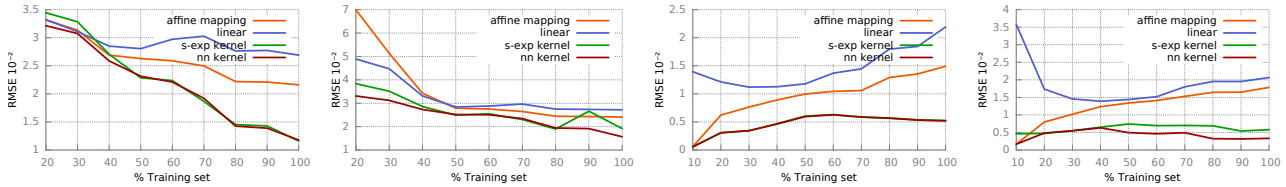
We presented in this work a novel solution for surface motion capture transfer that doesn't require source and target mesh correspondence. We contributed a Gaussian Process regression model applied directly to mesh data, and a correspondence densification scheme based on probabilistic dynamic time warping. Densifying the correspondences helps better capture motion style and variability. Concordantly, this richer variability is modelled more accurately with our non-linear regression. We also propose a user friendly algorithm for body parts separation and automatic stitching, and an example-based approach to improve predicted meshes with respect to the training set. This last heuristic compensates for the limitations of the linear parametrization of shape poses, which is mainly motivated by compatibility with standard regression kernels. A more elaborate regression model designed specifically for manifold valued data [2] could be attempted as a next step. We could also consider transferring appearance information along with geometry.

References

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

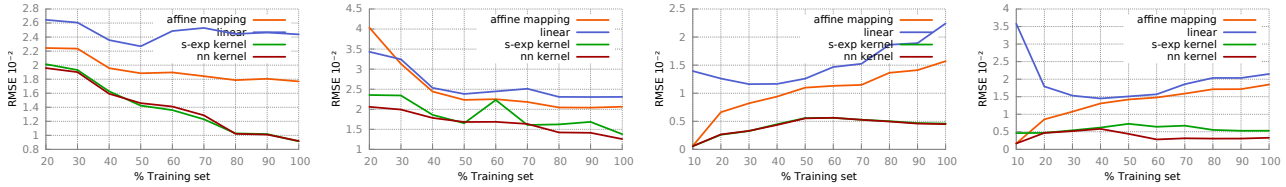
[2] M. Banerjee, R. Chakraborty, E. Ofori, M. S. Okun, D. E. Viallancourt, and B. C. Vemuri. A nonlinear regression technique for manifold valued data with applications to medical image analysis. In *CVPR*, 2016. 7

[3] I. Baran, D. Vlasic, E. Grinspun, and J. Popović. Semantic deformation transfer. In *ACM SIGGRAPH*, 2009. 1, 2, 3, 4, 7



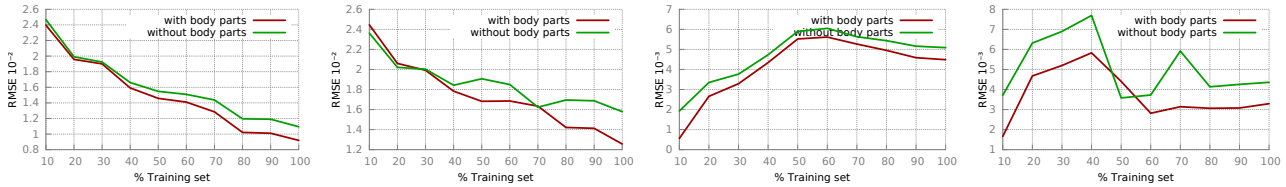
(a) Cross-val error, DAN to THOMAS (b) Cross-val error, THOMAS to DAN (c) Training error, DAN to THOMAS (d) Training error, THOMAS to DAN

Figure 7: Learning curves for various regression models.



(a) Cross-val error, DAN to THOMAS (b) Cross-val error, THOMAS to DAN (c) Training error, DAN to THOMAS (d) Training error, THOMAS to DAN

Figure 8: Learning curves for various regression models. Training subset initialized with key-poses.



(a) Cross-val error, DAN to THOMAS (b) Cross-val error, THOMAS to DAN (c) Training error, DAN to THOMAS (d) Training error, THOMAS to DAN

Figure 9: Learning curves for the GPR model with neural network kernel with and without body parts.

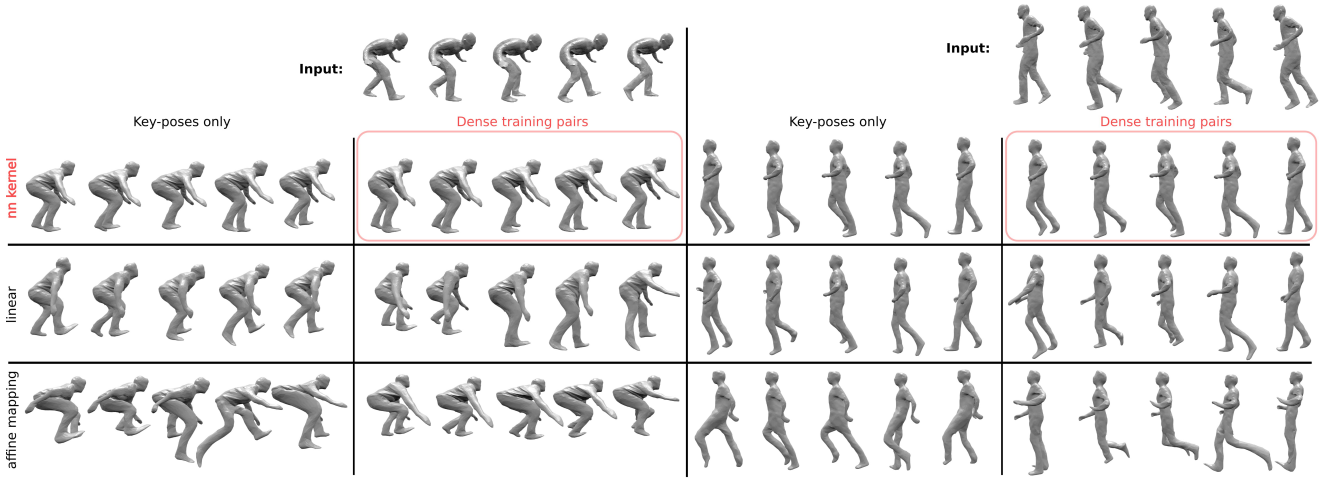


Figure 10: Transferring sequences *Duck* and *Run2walk* from THOMAS to DAN.

[4] M. Ben-Chen, O. Weber, and C. Gotsman. Spatial deformation transfer. In *SCA*, 2009. 2

[5] A. Boukhayma and E. Boyer. Video based Animation Synthesis with the Essential Graph. In *3DV*, 2015. 1, 6

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

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

[8] D. Casas, M. Volino, J. Collomosse, and A. Hilton. 4d video textures for interactive character appearance. *Computer Graphics Forum (Proceedings Eurographics)*, 33(2), 2014. 1, 6

[9] 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 Trans. Graph.*, 34(4), 2015. 1

[10] 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 Trans. Graph.*, 35(4), 2016. 1

[11] W.-W. Feng, B.-U. Kim, and Y. Yu. Real-time data driven deformation using kernel canonical correlation analysis. *ACM Trans. Graph.*, 27(3), 2008. 1, 2

[12] X. Huang, H. Fu, O. K.-C. Au, and C.-L. Tai. Optimal boundaries for poisson mesh merging. In *SPM*, 2007. 6

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

[14] M. Müller. *Information Retrieval for Music and Motion*. Springer-Verlag New York, Inc., 2007. 5

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

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

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

[18] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. 5

[19] H. Rhodin, J. Tompkin, K. I. Kim, V. Kiran, H.-P. Seidel, and C. Theobalt. Interactive motion mapping for real-time character control. *Computer Graphics Forum (Proceedings Eurographics)*, 33(2), 2014. 1, 2

[20] R. W. Sumner and J. Popović. Deformation transfer for triangle meshes. In *ACM SIGGRAPH*, 2004. 1, 2

[21] M. Tejera and A. Hilton. Learning part-based models for animation from surface motion capture. In *3DV*, 2013. 3

[22] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *CVPR*, 2016. 2

[23] A. Vögele, M. Hermann, B. Krüger, and R. Klein. Interactive steering of mesh animations. In *SCA*, 2012. 1, 2

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

[25] J. M. Wang, D. J. Fleet, and A. Hertzmann. Multifactor gaussian process models for style-content separation. In *ICML*, 2007. 2

[26] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30, 2008. 2

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

[28] D. Xu, H. Zhang, Q. Wang, and H. Bao. Poisson shape interpolation. In *SPM*, 2005. 6

[29] K. Yamane, Y. Ariki, and J. Hodgins. Animating Non-Humanoid Characters with Human Motion Data. In *SCA*, 2010. 1, 2

[30] 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 *ACM SIGGRAPH*, 2004. 6

[31] K. Zhou, W. Xu, Y. Tong, and M. Desbrun. Deformation transfer to multi-component objects. *Computer Graphics Forum*, 29(2), 2010. 2