Semi-Supervised Facial Animation Retargeting

Sofien Bouaziz* EPFL
Mark Pauly† EPFL

Figure 1: *Our facial animation retargeting system learns a mapping from motion capture data to arbitrary character parameters.*

Abstract

This paper presents a system for facial animation retargeting that allows learning a high-quality mapping between motion capture data and arbitrary target characters. We address one of the main challenges of existing example-based retargeting methods, the need for a large number of accurate training examples to define the correspondence between source and target expression spaces. We show that this number can be significantly reduced by leveraging the information contained in *unlabeled* data, i.e. facial expressions in the source or target space without corresponding poses. In contrast to labeled samples that require time-consuming and error-prone manual character posing, unlabeled samples are easily obtained as frames of motion capture recordings or existing animations of the target character. Our system exploits this information by learning a shared latent space between motion capture and character parameters in a semi-supervised manner. We show that this approach is resilient to noisy input and missing data and significantly improves retargeting accuracy. To demonstrate its applicability, we integrate our algorithm in a performance-driven facial animation system.

1 Introduction

Creating realistic facial animations is a complex task that usually requires a significant time commitment of highly skilled animators. Recent developments in facial motion capture systems allow speeding up this process by accurately capturing the performance of an actor, thereby shifting the complexity of facial animation towards retargeting. However, mapping the captured performance onto a virtual avatar is a highly non-trivial task, especially when the target character is not a close digital replica of the actor, as for example in the movie *King-Kong*. Low-level automatic methods are bound to fail, since establishing the correspondence between facial expressions of largely different characters requires high-level semantic knowledge of their expressions spaces. A common strategy is thus to provide a set of explicit point correspondences between these two spaces. For example, for a given recorded smile of the actor, an animator would create a semantically matching smile of the virtual target character. Given a set of such labeled pairs, retargeting essentially becomes a problem of scattered data approximation, i.e., extrapolating the explicit correspondences into the entire expression space. The main difficulty in this type of example-based retargeting is creating the examples. Typically a large number of correspondences needs to be established to adequately capture the subtleties of facial expressions. In addition, posing a character to match a recorded expression can be very difficult, as subtle motions, e.g. a slight raise of the eyebrows, are often overlooked. These minor inaccuracies can quickly lead to noticeable disturbances in the animations of the target character.

Contribution. In this paper, we present a novel example-based retargeting approach that significantly reduces the number of required training examples. Our method learns a shared latent space between motion capture and character parameters to represent their underlying common structure. Given a small set of manually specified correspondences between actor performance and target character expressions, the latent space is learned in a semi-supervised manner by using these labeled key poses, as well as the complete actor performance and previous animations of the target character. By adding this additional information we can increase the learning accuracy and stability, while the number of required training examples is reduced. We demonstrate that our system is resilient to noise and missing data, and can deal with high dimensional representations common in production-level facial rigs.

Related Work. Practical acquisition and motion capture systems have recently become robust, accurate, and affordable [Bradley et al. 2010; Beeler et al. 2011; Weise et al. 2011; Bouaziz et al. 2013] leading to a wider usage in professional and semi-professional productions. Since the seminal work of Williams [1990], numerous methods have been devoted to facial animation retargeting [Pighin and Lewis 2006]. Among those methods, approaches based on correspondences between motion capture markers and target characters [Bickel et al. 2007; Ma et al. 2008; Seol et al. 2012] have been successful when the actor and the animated faces are geometrically similar. Related to those approaches, [Noh and Neumann 2001; Sumner and Popović 2004] use dense correspondences between a source and a target mesh in order to retarget facial expression using vertex or triangle motion transfer. Numerous facial tracking and retargeting systems [Huang et al. 2011; Weise et al. 2011; Seol et al. 2012] use a blendshape representation based on Ekman's Facial Action Coding System [Ekman and Friesen 1978]. However, because of the linearity of the blendshape model, reproducing subtle non-linear motion is difficult.

Our system is most closely related to example-based methods [Deng et al. 2006; Song et al. 2011; Kholgade et al. 2011] that do not require any similarity between the source and the target face. The main difference to existing solutions is that our approach

*e-mail:sofien.bouaziz@epfl.ch
†e-mail:mark.pauly@epfl.ch

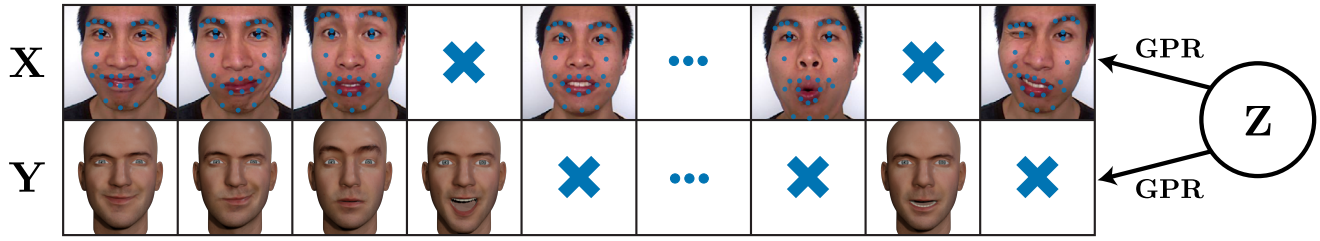

Figure 2: *Our algorithm learns a shared latent space* $\mathbf{Z}$ *from a space* $\mathbf{X}$ *of motion capture parameters and a space* $\mathbf{Y}$ *of character parameters. Gaussian Process Regressors (GPR) are used to model the mappings from the latent space onto the observation spaces. In order to train the GPRs only few pairwise correspondences between* $\mathbf{X}$ *and* $\mathbf{Y}$ *need to be specified. A key feature of our algorithm is that we also incorporate unlabeled data points for which no correspondence is given.*

supports non-linear retargeting of motion capture data and exploits unlabeled data to improve the retargeting accuracy with a reduced number of training examples.

The core of our facial animation retargeting system is based on recent works on Gaussian Process Latent Variable Models (GPLVM) [Lawrence 2004]. GPLVM was used successfully for human body tracking [Urtasun et al. 2006], retargeting [Yamane et al. 2010] and inverse kinematics [Grochow et al. 2004]. Recently, GPLVM has been extended to support multiple observation spaces [Ek 2009], missing data [Navaratnam et al. 2007] and constrains over the latent space [Urtasun et al. 2007; Wang et al. 2008]. In our work we enhance the shared GPLVM [Ek 2009] with a prior over latent configurations allowing to preserve local distances of the observation spaces. This prior takes its roots in manifold alignment [Ham et al. 2005] and Gaussian random fields [Zhu et al. 2003; Verbeek and Vlassis 2006].

2 Learning

Classical example-based retargeting establishes a mapping from the source to the target space by computing an interpolation function from the point-wise correspondences defined by the labeled examples. Our method is based on one key observation: *unlabeled* frames can provide valuable information to establish this mapping. With unlabeled frames we mean poses in the captured sequence for which no corresponding expression for the target has been specified. For motion capture data, these unlabeled data points are abundant, since typically many hundreds of frames are recorded and only few are manually labelled. The main advantage of incorporating unlabeled data is that they provide important information about the local structure of the expressions space, which leads to better alignment of source and target spaces when computing the mapping. We can even go further and also incorporate unlabeled expressions of the target character, which help to constrain the mapping function by defining the space of semantically correct expressions of the target. Unlabeled target character samples are often available in the form of pre-existing animations that, for example, have been generated by an artist.

We employ shared GPLVM [Ek 2009] to learn a mapping between motion capture and character parameters. The main hypothesis here is that both parameter spaces are (non-linearly) generated from a common low-dimensional manifold. Shared GPLVM (sGPLVM) learns a shared latent space by training Gaussian Process Regressors (GPR) to model the generative mappings from the latent space onto the observation spaces as illustrated in Figure 2. Gaussian Process Regressors can be trained robustly from small training sets and their parameters can be learned by maximizing the marginal likelihood of the training data. This is more efficient than techniques that use cross-validation to infer the parameter values when the training set is small, since the training dataset does not need to be reduced further [Rasmussen and Williams 2006].

2.1 Shared GPLVM Learning

Assume we are given two sets of corresponding observations $\mathbf{X} = \left[\mathbf{x}_1, \ldots, \mathbf{x}_n\right]^T$ and $\mathbf{Y} = \left[\mathbf{y}_1, \ldots, \mathbf{y}_n\right]^T$, where $\mathbf{x}_i \in \mathbb{R}^{d_x}$ and $\mathbf{y}_i \in \mathbb{R}^{d_y}$. In our retargeting system $\mathbf{X}$ represents the space of source motion capture parameters and $\mathbf{Y}$ the space of target virtual character parameters. Let $\mathbf{Z} = \left[\mathbf{z}_i, \ldots, \mathbf{z}_n\right]^T$, $\mathbf{z}_i \in \mathbb{R}^{d_z}$ denote the corresponding (unknown) shared latent points. We model the generative mapping from the latent space onto the observation spaces with Gaussian processes using the conditional probabilities

$$P(\mathbf{X}|\mathbf{Z}) = \frac{1}{\sqrt{2\pi^{nd_x}|\mathbf{K}_{\mathbf{Z},\mathbf{\Phi}_X}|^{d_x}}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{K}_{\mathbf{Z},\mathbf{\Phi}_X}^{-1}\mathbf{X}\mathbf{X}^T\right)\right), \tag{1}$$

$$P(\mathbf{Y}|\mathbf{Z}) = \frac{1}{\sqrt{2\pi^{nd_y}|\mathbf{K}_{\mathbf{Z},\mathbf{\Phi}_Y}|^{d_y}}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{K}_{\mathbf{Z},\mathbf{\Phi}_Y}^{-1}\mathbf{Y}\mathbf{Y}^T\right)\right). \tag{2}$$

The vector $\mathbf{\Phi} = \{\theta_1, \theta_2, \theta_3\}$ defines the parameters of the kernel $\mathbf{K}_{\mathbf{Z},\mathbf{\Phi}}$ given as

$$\mathbf{K}_{\mathbf{Z},\mathbf{\Phi}}^{i,j} = k_{\mathbf{\Phi}}(\mathbf{z}_i, \mathbf{z}_j) = \theta_1 \exp\left(-\frac{\theta_2}{2}||\mathbf{z}_i - \mathbf{z}_j||_2^2\right) + \theta_3^{-1}\delta_{i,j}, \tag{3}$$

where $\mathbf{K}_{\mathbf{Z},\mathbf{\Phi}}^{i,j}$ is the element located at the i-th line and j-th column of the kernel matrix $\mathbf{K}_{\mathbf{Z},\mathbf{\Phi}}$ and $\delta_{i,j}$ is the Kronecker delta. Learning a shared GPLVM amounts to estimating the latent positions and kernel parameters by maximizing

$$\underset{\mathbf{Z},\mathbf{\Phi}_X,\mathbf{\Phi}_Y}{\arg\max}\, P(\mathbf{Z}|\mathbf{X},\mathbf{Y}) = \underset{\mathbf{Z},\mathbf{\Phi}_X,\mathbf{\Phi}_Y}{\arg\max}\, P(\mathbf{X}|\mathbf{Z})P(\mathbf{Y}|\mathbf{Z})P(\mathbf{Z}). \tag{4}$$

Semi-supervised learning. An important benefit of the shared GPLVM is that it can directly incorporate extra data points that do not need to be in correspondence. We can thus learn the shared GPLVM using $\mathbf{X} = \left[\mathbf{X}_l^T, \mathbf{X}_u^T, \circ\right]^T$ and $\mathbf{Y} = \left[\mathbf{Y}_l^T, \circ, \mathbf{Y}_u^T\right]^T$, where labeled pairs are denoted by $\mathbf{X}_l \in \mathbb{R}^{l \times d_x}$ and $\mathbf{Y}_l \in \mathbb{R}^{l \times d_y}$, and unlabeled samples are given by $\mathbf{X}_u \in \mathbb{R}^{m \times d_x}$, $\mathbf{Y}_u \in \mathbb{R}^{n \times d_y}$ with the $\circ$ indicating the missing correspondences (see Figure 2).

By using smooth mappings from the latent space to the observation spaces, sGPLVM ensures that close points in the latent space remain close in the observation spaces. However, the inverse is not

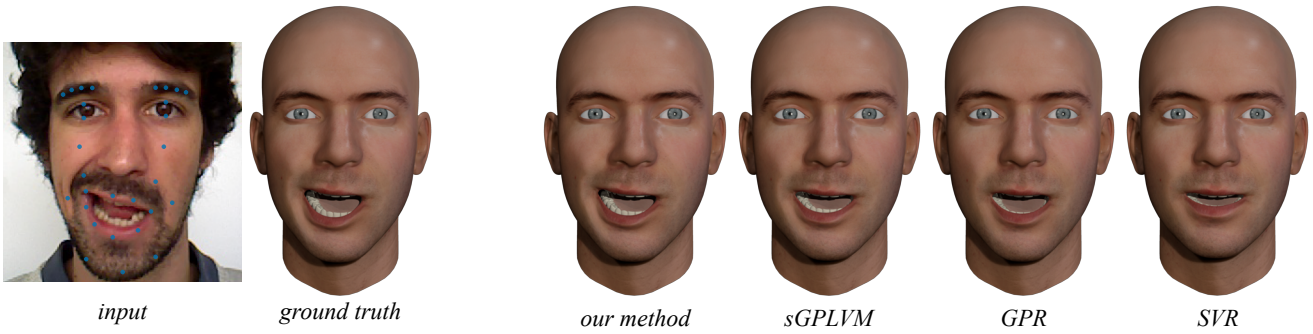

Figure 3: *Our method retargets accurately the facial expressions of the actor. With a small number of labels SVR has tendency to damp the facial expressions. In our examples, GPR gives results similar or slightly less accurate than sGPLVM, which we further improve in our method by incorporating unlabeled data.*

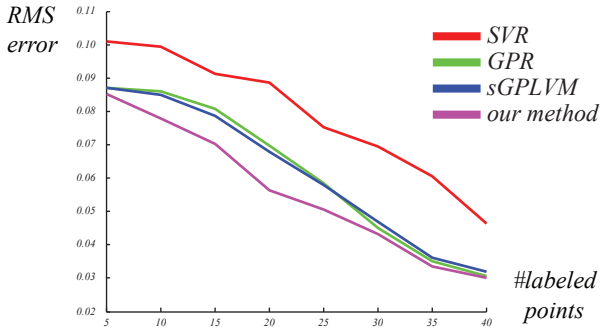

Figure 4: *A quantitative comparison of different learning approaches shows the root mean square (RMS) distance to the ground truth as a function of the number of training examples.*

necessarily true, i.e., points close in the observation spaces may be far apart in the latent space. In order to preserve the local topological structure of $\mathbf{X}$ and $\mathbf{Y}$ in the latent space, we therefore define a prior based on local linear embedding (LLE) [Roweis and Saul 2000] over the latent configurations. LLE assumes that each data point of the observation spaces and its neighbors are close to a locally linear patch on the manifold. The local geometry of these patches can then be encoded by linear coefficients w_{ij} that reconstruct each data point from its neighbors. By enforcing that the reconstruction of each latent point from its neighbors follows the same set of coefficients than their corresponding high dimensional point, the local structure of the observation spaces can be preserved in the latent space. We model this concept with a prior over the latent configuration using a Gaussian process

$$P(\mathbf{Z}) = \frac{1}{\sqrt{2\pi^{(l+m+n)d_z}|\mathbf{L}^{-1}|^{d_z}}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{L}\mathbf{Z}\mathbf{Z}^T\right)\right), \quad (5)$$

where $\mathbf{L} = \mathbf{M}^T\mathbf{M} + \mathbf{I}$ and $\mathbf{M}$ is a matrix in which each line encodes one reconstruction constraint and is defined as

$$\mathbf{M} = \begin{bmatrix} (\mathbf{I}-\mathbf{C}_X)^{:,1:l} & (\mathbf{I}-\mathbf{C}_X)^{:,(l+1):(l+m)} & 0 \\ (\mathbf{I}-\mathbf{C}_Y)^{:,1:l} & 0 & (\mathbf{I}-\mathbf{C}_Y)^{:,(l+1):(l+n)} \end{bmatrix}. \quad (6)$$

In the formulation above, $\mathbf{A}^{:,i:j}$ denotes a block a the matrix $\mathbf{A}$ going from column i to column j and

$$\mathbf{C}_U^{i,j} = c_U(\mathbf{u}_i, \mathbf{u}_j) = \begin{cases} w_{ij} \text{ if } j \in N_i, \\ 0 \text{ otherwise.} \end{cases} \quad (7)$$

N_i are the indices of the k-nearest neighbor of $\mathbf{u}_i$ and the coefficients w_{ij} are defined as

$$w_{ij} = \arg\min_{w_{ij}} ||\mathbf{u}_i - \sum_{j \in N_i} w_{ij}\mathbf{u}_j||_2^2 \quad \text{s.t.} \sum_{j \in N_i} w_{ij} = 1. \quad (8)$$

Incorporating this prior of the local structure of the observation spaces helps to better constrain the position of the points with missing correspondences in the latent space. We also found that it helps increase the robustness of the training to bad initialization of the latent coordinates.

2.2 Computing the Mapping Function

The mapping from motion capture parameters to character parameters is done in two steps. We first solve for the latent position $\mathbf{z}_k^*$ given the motion capture observation $\tilde{\mathbf{x}}_k$. We call this part *source mapping*. Given the latent position $\mathbf{z}^*$, the subsequent *target mapping* part solves for the character parameters $\mathbf{y}^*$.

Source mapping. The source mapping not only solves for the latent position $\mathbf{z}_k^*$, but also for the most likely capture parameters $\mathbf{x}_k^*$ given the observation $\tilde{\mathbf{x}}_k$, the optimized motion capture parameters $\mathbf{x}_{k-1}^*$ of the previous frame, and the training data $\mathbf{X}$ and $\mathbf{Z}$. Thus we optimize

$$\arg\max_{\mathbf{x}_k^*, \mathbf{z}_k^*} P(\mathbf{x}_k^*, \mathbf{z}_k^* | \mathbf{x}_{k-1}^*, \tilde{\mathbf{x}}_k, \mathbf{X}, \mathbf{Z}). \quad (9)$$

We approximate the above probability density function by assuming that $\mathbf{z}_k^*$ is independent of $\mathbf{x}_{k-1}^*$, $\tilde{\mathbf{x}}_k$, $\mathbf{X}$, and $\mathbf{Z}$. This allows us to reformulate the optimization as

$$\arg\max_{\mathbf{x}_k^*, \mathbf{z}_k^*} P(\mathbf{x}_k^* | \mathbf{z}_k^*, \mathbf{x}_{k-1}^*, \tilde{\mathbf{x}}_k, \mathbf{X}, \mathbf{Z}) P(\mathbf{z}_k^*), \quad (10)$$

which can be extended to

$$\arg\max_{\mathbf{x}_k^*, \mathbf{z}_k^*} P(\mathbf{x}_k^*, \mathbf{x}_{k-1}^*, \tilde{\mathbf{x}}_k | \mathbf{z}_k^*, \mathbf{X}, \mathbf{Z}) P(\mathbf{z}_k^*). \quad (11)$$

By further assuming that $\tilde{\mathbf{x}}_k$ and $\mathbf{x}_{k-1}^*$ are independent of $\mathbf{z}_k^*$, $\mathbf{X}$, and $\mathbf{Z}$ given $\mathbf{x}_k^*$, and $\tilde{\mathbf{x}}_k$ is independent of $\mathbf{x}_{k-1}^*$ given $\mathbf{x}_k^*$, we obtain our final optimization objective

$$\arg\max_{\mathbf{x}_k^*, \mathbf{z}_k^*} P(\mathbf{x}_k^*|\mathbf{z}_k^*, \mathbf{X}, \mathbf{Z}) P(\tilde{\mathbf{x}}_k|\mathbf{x}_k^*) P(\mathbf{x}_{k-1}^*|\mathbf{x}_k^*) P(\mathbf{z}_k^*). \quad (12)$$

The likelihoods $P(\tilde{\mathbf{x}}_k|\mathbf{x}_k^*)$ and $P(\mathbf{x}_{k-1}^*|\mathbf{x}_k^*)$ represent closeness to the observation and temporal smoothness, respectively, and are modeled by two Gaussian distributions as

$$P(\tilde{\mathbf{x}}_k|\mathbf{x}_k^*) = \mathcal{N}(\tilde{\mathbf{x}}_k|\mathbf{x}_k^*, \sigma_c^2\mathbf{I}), \quad (13)$$

$$P(\mathbf{x}_{k-1}^*|\mathbf{x}_k^*) = \mathcal{N}(\mathbf{x}_{k-1}^*|\mathbf{x}_k^*, \sigma_t^2\mathbf{I}). \quad (14)$$

The two probabilities $P(\mathbf{x}_k^*|\mathbf{z}_k^*, \mathbf{X}, \mathbf{Z})$ and $P(\mathbf{z}_k^*)$ act as priors over motion capture parameters and latent position and are defined as

$$P(\mathbf{x}_k^*|\mathbf{z}_k^*, \mathbf{X}, \mathbf{Z}) = \mathcal{N}(\mathbf{x}_k^*|\boldsymbol{\mu}, \sigma_p^2\mathbf{I}), \quad (15)$$

$$\boldsymbol{\mu} = \mathbf{K}_{\mathbf{Z},\boldsymbol{\Phi}_X}^{-1}\mathbf{X}\mathbf{k}_{\boldsymbol{\Phi}_X}(\mathbf{z}_k^*), \quad (16)$$

$$\sigma_p^2 = k_{\boldsymbol{\Phi}_X}(\mathbf{z}_k^*, \mathbf{z}_k^*) - \mathbf{k}_{\boldsymbol{\Phi}_X}(\mathbf{z}_k^*)^T\mathbf{K}_{\mathbf{Z},\boldsymbol{\Phi}_X}^{-1}\mathbf{k}_{\boldsymbol{\Phi}_X}(\mathbf{z}_k^*), \quad (17)$$

where $\mathbf{k}_{\boldsymbol{\Phi}}(\mathbf{z}_k^*)$ is a vector whose i-th element is $k_{\boldsymbol{\Phi}}(\mathbf{z}_k^*, \mathbf{z}_i)$ and $P(\mathbf{z}_k^*) = \mathcal{N}(\mathbf{z}_k^*|0, \mathbf{I})$. One advantage of this formulation is that missing dimensions of $\tilde{\mathbf{x}}_k$ can be retrieved during the optimization by setting $\sigma_c^2 = \infty$ in Equation 13 for these dimensions.

Target mapping. The second step of the mapping process is to find the character parameters $\mathbf{y}^*$ given the latent position $\mathbf{z}^*$ by maximizing

$$\arg\max_{\mathbf{y}_k^*} P(\mathbf{y}_k^*|\mathbf{z}_k^*, \mathbf{Y}, \mathbf{Z}) = \mathbf{K}_{\mathbf{Z},\boldsymbol{\Phi}_Y}^{-1}\mathbf{Y}\mathbf{k}_{\boldsymbol{\Phi}_Y}(\mathbf{z}_k^*). \quad (18)$$

Implementation. In our implementation, we first mean center the observation spaces and rescale them by dividing by their maximum variance. For the learning phase, we empirically found $\boldsymbol{\Phi} = \{1, 1, 100\}$ to be good initial kernel parameters for the optimization for all our examples. We fix σ_c^2 and σ_t^2 by estimating the noise level of the motion capture system [Weise et al. 2011] and chose $k = 8$ nearest neighbors for LLE and 8 dimensions for the latent space. The latent coordinates are initialized using the semi-supervised manifold alignment technique presented in [Ham et al. 2005]. For the mapping phase, we initialize $\mathbf{x}_k^*$ with the motion capture observation $\tilde{\mathbf{x}}_k$ and $\mathbf{z}_k^*$ with the latent position corresponding to the closest $\mathbf{x}_i$ to $\tilde{\mathbf{x}}_k$. We use scaled conjugate gradient [Moller 1993] as optimizer and minimize the negative logarithm of the probabilities.

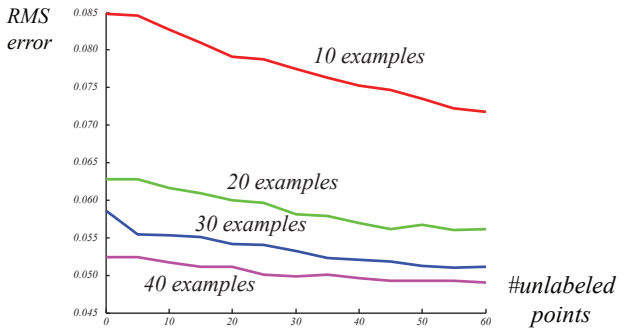

Figure 5: *Unlabeled data points help to increase retargeting accuracy, in particular when working with few training examples.*

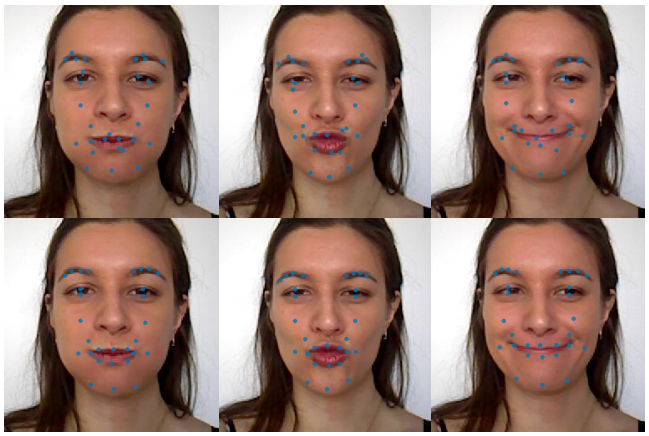

Figure 6: *Resilience to noise. Our learning approach is able to compute accurate marker positions (bottom row) by automatically correcting the noisy input points (top row).*

3 Evaluation

For our evaluation experiments, we use the *faceshift* tracking system (www.faceshift.com). Given a recorded sequence of a human actor, this system produces an animated 3D mesh represented in a blendshape basis that matches the actor's performance. We select a set of vertices on the mesh as marker positions to generate motion capture input and perform a retargeting of these marker points onto the blendshape basis of the animated target character. This setup allows measuring and comparing the performance of our algorithm since the blendshape parameters provided by the tracking system can be treated as ground truth for the evaluation. Note that all other retargeting sequences use target characters (the models shown in Figure 1, see also video) for which no such ground truth data is available.

Comparison. We compare our algorithm with Support Vector Regression (SVR) [Drucker et al. 1996], Gaussian Process Regression (GPR) [Williams and Rasmussen 1995] and the supervised shared GPLVM (sGPLVM) [Ek 2009]. We recorded sequences of approximatively 2000 frames of different actors. The different algorithms are applied 20 times over those sequences by random selection of labelled and unlabeled points, using 100 unlabeled data points for both observation spaces. The averaged results shown in Figures 3 and 4 demonstrate that our algorithm improves the retargeting accuracy by up to 20%, especially when the number of labeled expression correspondences is small. As demonstrated in the accompanying video, our algorithm preserves motion dynamics significantly better than the other approaches.

Unlabeled points. Figure 5 illustrates the effect of using unlabeled points for establishing the retargeting mapping function. As the curves indicate, when using about 50 unlabeled points we can achieve the same retargeting accuracy with 20 training examples as with 30 examples and no additional unlabeled points. Compared to the time-consuming and error-prone labeling, the latter come essentially for free, allowing for significant savings in manual labor. Unlabeled points are particularly useful for small sets of manually specified examples as the given correspondences do not span the full animation space.

Noise and missing data. One advantage of our formulation is its robustness to noise (Figure 6) and missing data (Figure 7). Our

Figure 7: *Missing markers can be handled by our retargeting system. The optimization jointly retrieves the location of the missing markers (green) and the target character parameters.*

system models a probability distribution function over motion capture parameters and latent positions allowing to retrieve the most probable set of markers given the possible noisy or incomplete input observation.

Character posing. The resilience of our algorithm to missing data is not limited to the input space. We can exploit the regularization of our probabilistic framework to also complete missing data in the target space, which offers a simple but effective approach to character posing. The animator can specify only a subset of the target animation parameters and our algorithm will automatically infer the most probable pose matching the specified values (see Figure 8). This type of guided character posing is particularly advantageous for complex animation models, where many parameters only induce subtle pose variations that are thus difficult to specify, but nevertheless important for the expression.

Discussion and Limitations. When the number of examples is small, example-based retargeting methods have a tendency to infer a wrong correlation between parts of the face as for example mouth open and eyebrows up. This effect is reduced in our approach by taking into account unlabeled data. One additional solution is to split the face (e.g. upper part and lower part) and to learn the retargeting independently for those parts, similar to recent linear 3D face models [Tena et al. 2011].

In our work, we use a set of key poses, rather than sequences, to learn the retargeting function. Learning a latent dynamical system as in [Wang et al. 2008] with different motion style is challenging especially with a small set of sequences. Nevertheless, motion sequences can additionally be used in our approach by taking into account temporal closeness when building the matrix in Equation 6.

A drawback of the Gaussian Process Regressor model is its time complexity, which is $O(N^3)$ for the training phase and $O(N^2)$ for evaluating the mapping, where N is the number of points in the training data. Sparse approximations [Lawrence et al. 2003; Lawrence 2007] allow to reduce the training complexity to a more manageable $O(k^2N)$ where k is the number of active points retained in the sparse representation. In practice, our current implementation supports realtime retargeting for a training set of a few hundred data points for each observation space. The training time of our system for 40 examples and 100 unlabeled points is around 1-2 minutes and the mapping between 30 to 40ms.

In our current implementation the dimension of the latent space is chosen empirically. Recent works in non-linear dimensionality reduction [Geiger et al. 2009; Salzmann et al. 2010] introduced a rank prior that allows to automatically determine the dimension of the latent space. This work should also be applicable for our approach.

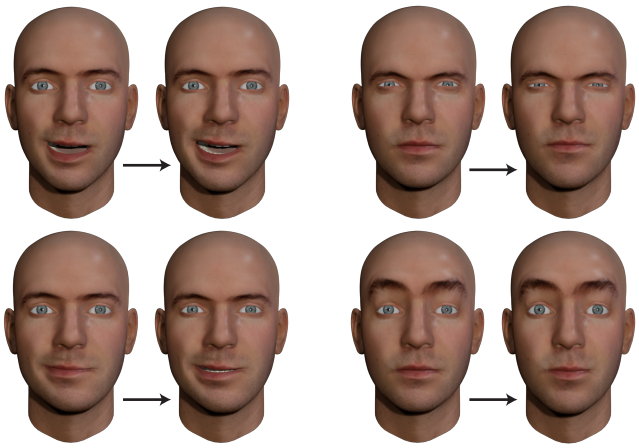

Figure 8: *Character posing can be simplified by optimizing for the missing animation parameters. In these examples, the animator only needs to specify 2-3 animation parameters (left) and the system automatically infers the most likely pose matching this input (right), activating about 20 additional blendshape parameters.*

4 Conclusion

We have introduced a novel statistical approach to high-quality facial animation retargeting that achieves better results than other non-linear regression techniques. By leveraging the information contained in unlabeled data, a key novelty in our retargeting approach, we can reduce the number of required training examples. We have shown that our approach is well suited to retargeting facial animations from motion capture data, as posing a character is time consuming, while unlabeled data is easily obtained by tracking the actor. Since our method implicitly learns a low-dimensional representation, our system has no difficulty dealing with complex, high-dimensional input or output data commonly used in studio productions. At the same time, the robustness of our approach to noise and missing data makes the method particularly suitable for low-cost motion capture systems. In addition, our method can simplify character posing by exploiting the correlation between the different character parameters.

We believe that the main features of our approach will be applicable in other retargeting applications and see several avenues for future research. A promising idea is to further explore manifold alignment algorithms [Yang et al. 2008; Wang and Mahadevan 2009; Zhai et al. 2010] to define a prior over latent configurations and for the initialization of the shared GPLVM. Our statistical framework is also well suited for active learning. We expect further improvements in retargeting accuracy when automatically suggesting new poses for labeling based on an online analysis of the uncertainty of the current retargeting mapping function.

Aknowlegments. We thank Mario Christoudias, Neil Lawrence, Raquel Urtasun, Mathieu Salzmann, and Andreas Damianou for their valuable comments, Brian Amberg, Thibaut Weise and faceshift AG (www.faceshift.com) for their help and support. We are grateful to Thibaut Weise, Minh Dang, Eva Darulova, Mario Deuss, Laura Gosmino, and Giuliano Losa for being great actors, and to all the other people who took part in the experiments. This research is supported by the Swiss National Science Foundation grant 20PA21L_129607.

References

BEELER, T., HAHN, F., BRADLEY, D., BICKEL, B., BEARDSLEY, P., GOTSMAN, C., SUMNER, R. W., AND GROSS, M. 2011. High-quality passive facial performance capture using anchor frames. *ACM Trans. Graph.*.

BICKEL, B., BOTSCH, M., ANGST, R., MATUSIK, W., OTADUY, M., PFISTER, H., AND GROSS, M. 2007. Multi-scale capture of facial geometry and motion. *ACM Trans. Graph.*.

BOUAZIZ, S., WANG, Y., AND PAULY, M. 2013. Online modeling for realtime facial animation. *ACM Trans. Graph.*.

BRADLEY, D., HEIDRICH, W., POPA, T., AND SHEFFER, A. 2010. High resolution passive facial performance capture. *ACM Trans. Graph.*.

DENG, Z., CHIANG, P.-Y., FOX, P., AND NEUMANN, U. 2006. Animating blendshape faces by cross-mapping motion capture data. In *I3D*.

DRUCKER, H., BURGES, C. J. C., KAUFMAN, L., SMOLA, A. J., AND VAPNIK, V. 1996. Support vector regression machines. In *NIPS*.

EK, C. 2009. *Shared Gaussian Process Latent Variable Models*. PhD thesis.

EKMAN, P., AND FRIESEN, W. 1978. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press.

GEIGER, A., URTASUN, R., AND DARRELL, T. 2009. Rank priors for continuous non-linear dimensionality reduction. In *CVPR*.

GROCHOW, K., MARTIN, S. L., HERTZMANN, A., AND POPOVIĆ, Z. 2004. Style-based inverse kinematics. *ACM Trans. Graph.*.

HAM, J. H., LEE, D. D., AND SAUL, L. K. 2005. Semisupervised alignment of manifolds. In *Proc. of the 10th International Workshop on Artificial Intelligence and Statistics*.

HUANG, H., CHAI, J., TONG, X., AND WU, H.-T. 2011. Leveraging motion capture and 3d scanning for high-fidelity facial performance acquisition. *ACM Trans. Graph.*.

KHOLGADE, N., MATTHEWS, I., AND SHEIKH, Y. 2011. Content retargeting using parameter-parallel facial layers. *SCA*.

LAWRENCE, N., SEEGER, M., AND HERBRICH, R. 2003. Fast sparse gaussian process methods: The informative vector machine. In *NIPS*.

LAWRENCE, N. D. 2004. Gaussian process latent variable models for visualisation of high dimensional data. In *NIPS*.

LAWRENCE, N. D. 2007. Learning for larger datasets with the gaussian process latent variable model. In *Proc. of the 11th Int. Workshop on Artificial Intelligence and Statistics*.

MA, W.-C., JONES, A., CHIANG, J.-Y., HAWKINS, T., FREDERIKSEN, S., PEERS, P., VUKOVIC, M., OUHYOUNG, M., AND DEBEVEC, P. 2008. Facial performance synthesis using deformation-driven polynomial displacement maps. In *Proc. of ACM SIGGRAPH Asia*.

MOLLER, M. F. 1993. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*.

NAVARATNAM, R., FITZGIBBON, A. W., AND CIPOLLA, R. 2007. The joint manifold model for semi-supervised multi-valued regression. In *ICCV*.

NOH, J.-Y., AND NEUMANN, U. 2001. Expression cloning. In *Proc. of ACM SIGGRAPH*.

PIGHIN, F., AND LEWIS, J. P. 2006. Facial motion retargeting. In *ACM SIGGRAPH Courses*.

RASMUSSEN, C. E., AND WILLIAMS, C. 2006. *Gaussian Processes for Machine Learning*. MIT Press.

ROWEIS, S. T., AND SAUL, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science*.

SALZMANN, M., EK, C. H., URTASUN, R., AND DARRELL, T. 2010. Factorized orthogonal latent spaces. *Journal of Machine Learning Research*.

SEOL, Y., LEWIS, J., SEO, J., CHOI, B., ANJYO, K., AND NOH, J. 2012. Spacetime expression cloning for blendshapes. *ACM Trans. Graph.*.

SONG, J., CHOI, B., SEOL, Y., AND NOH, J. 2011. Characteristic facial retargeting. *Computer Animation and Virtual Worlds*.

SUMNER, R. W., AND POPOVIĆ, J. 2004. Deformation transfer for triangle meshes. *ACM Trans. Graph.*.

TENA, J. R., TORRE, F. D. L., AND MATTHEWS, I. 2011. Interactive region-based linear 3d face models. *ACM Trans. Graph.*.

URTASUN, R., FLEET, D. J., AND FUA, P. 2006. 3d people tracking with gaussian process dynamical models. In *CVPR*.

URTASUN, R., FLEET, D. J., AND LAWRENCE, N. D. 2007. Modeling human locomotion with topologically constrained latent variable models. In *Proc. 2nd Conf. on Human Motion*.

VERBEEK, J. J., AND VLASSIS, N. 2006. Gaussian fields for semi-supervised regression and correspondence learning. *Pattern Recogn.*.

WANG, C., AND MAHADEVAN, S. 2009. A general framework for manifold alignment. In *AAAI Symposium on Manifold Learning and its Applications*.

WANG, J. M., FLEET, D. J., AND HERTZMANN, A. 2008. Gaussian process dynamical models for human motion. *PAMI*.

WEISE, T., BOUAZIZ, S., LI, H., AND PAULY, M. 2011. Realtime performance-based facial animation. *ACM Trans. Graph.*.

WILLIAMS, C. K. I., AND RASMUSSEN, C. E. 1995. Gaussian processes for regression. In *NIPS*.

WILLIAMS, L. 1990. Performance-driven facial animation. *Computer Graphics (Proceedings of SIGGRAPH)*.

YAMANE, K., ARIKI, Y., AND HODGINS, J. 2010. Animating non-humanoid characters with human motion data. In *SCA*.

YANG, G., XU, X., AND ZHANG, J. 2008. Manifold alignment via local tangent space alignment. In *Proc. of Int. Conf. on Comp Sc. and Soft. Eng.*

ZHAI, D., LI, B., CHANG, H., SHAN, S., CHEN, X., AND GAO, W. 2010. Manifold alignment via corresponding projections. In *BMVC*.

ZHU, X., LAFFERTY, J., AND GHAHRAMANI, Z. 2003. Semi-supervised learning: From gaussian fields to gaussian processes. Tech. rep., School of CS, CMU.