Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2009)
E. Grinspun and J. Hodgins (Editors)

Style Learning and Transferring for Facial Animation Editing

Xiaohan Ma, Binh Huy Le, and Zhigang Deng

Computer Graphics and Interactive Media Lab
Department of Computer Science
University of Houston, Houston, TX
Emails: {xiaohan|bhle2|zdeng}@cs.uh.edu

Abstract

Most of current facial animation editing techniques are frame-based approaches (i.e., manually edit one keyframe every several frames), which is ineffective, time-consuming, and prone to editing inconsistency. In this paper, we present a novel facial editing style learning framework that is able to learn a constraint-based Gaussian Process model from a small number of facial-editing pairs, and then it can be effectively applied to automate the editing of the remaining facial animation frames or transfer editing styles between different animation sequences. Comparing with the state of the art, multiresolution-based mesh sequence editing technique, our approach is more flexible, powerful, and adaptive. Our approach can dramatically reduce the manual efforts required by most of current facial animation editing approaches.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism —- Animation; Artificial Intelligence [I.2.6]: Learning —- Analogies

1. Introduction

Providing users efficient and intuitive facial animation editing tools has been a long-standing challenging problem to the computer animation community. Most of current facial animation editing approaches [JTDP03, ZSCS04, BLB*08, FKY08] need users to sculpt a facial keyframe every several frames. The users often need to repeat the similar process for different animation sequences or models, which is painstakingly time-consuming. Another common limitation of these frame-based editing methods is that it is very difficult to ensure the consistency of editing styles across frames or among different animation sequences. Therefore, it raises two interesting questions: 1) Is it possible to learn the “editing style" from a small set of facial editing pairs or operations and then automatically transfer it to other frames of the same sequence or other facial animation sequences? And 2) how can we incorporate meaningful constraints to the above facial animation editing-style learning process?

Motivated by the above questions, in this paper we present a novel style learning and transferring framework for facial animation editing. It is able to adaptively learn the editing style from a small set of facial-editing pairs (*i.e.*, selected facial frames before/after the editing) and then automatically apply it to other unedited frames or facial animation sequences. Meanwhile, its algorithm can ensure the motion smoothness, editing consistency, and the satisfaction of user-specified facial constraints via a introduced Constraint-based Gaussian Process (CGP) model and a multi-level facial constraint model. By comparing our approach with the state of the art mesh sequence editing techniques, we show that our approach is more flexible, powerful, and adaptive.

The main contribution of this work includes: (1) The introduced Constraint-based Gaussian Process model (CGP) can effectively learn the editing style from a small set of facial editing pairs, and then it can be effectively applied to automate the editing of the remaining facial animation frames or transfer editing styles between different animation sequences. (2) The introduced multi-level facial constraint model provides users a flexible and powerful control mechanism. The combination of our context-dependent CGP model and multi-level facial constraints is able to dramatically reduce the painstakingly manual efforts involved in most of current facial animation editing practices, *e.g.*, sculpting a facial keyframe every several frames.

2. Related Work

Facial expression editing: Early data-driven facial expression editing work focused on segmenting the face into different regions and then performing editing operations [JTDP03, ZLGS03, ZSCS04, LD08]. Cao *et al.* [CFP03] employ Independent Component Analysis (ICA) to decompose facial motion signals into emotional and speech components, and then perform various editing operations on different ICA components. Recently, Sucontphunt *et al.* [SMND08] developed a 2D portrait-based facial expression posing interface, and user-edited 2D portraits are automatically mapped to the best matched 3D facial expressions in a dataset. Bickel *et al.* [BLB*08] presented a hybrid editing technique that combines large-scale deformation with fine-scale facial details such as wrinkles. Feng *et al.* [FKY08] presented a data-driven geometric deformation framework based on the kernel Canonical Correlation Analysis algorithm (CCA). They showed their approach can be effectively used for facial expression editing.

The above facial expression editing approaches are essentially frame-based techniques, and thus animators have to artistically ensure animation smoothness and style consistency across sequences, which is often a daunting task even for skilled animators. Different from these frame-based approaches, our approach employs a different paradigm: taking a small set of facial-editing pairs (selected facial frames before/after the editing) as the training set and learning a statistical model to encode the “editing style", and then, the editing style model can be used to automate the editing of other frames or sequences. Furthermore, our approach automatically ensures style consistency across sequences.

Surface animation editing: Kircher and Garland [KG05, KG06] apply time-varying multiresolution transformation algorithms to modify arbitrarily deforming surface animations in a temporarily coherent manner. They further extend it to perform rotational editing for free-from motions [KG08]. Xu *et al.* [XZY*07] proposed a keyframe-based mesh sequence editing approach where selected keyframes are adjusted to satisfy user-specified constraints and then the deformation is propagated to the whole sequence. However, this approach requires users to manually edit a substantial number of keyframes, and it is nontrivial to construct proper local tangent planes required in their approach. Sumner *et al.* [SSP07] construct a deformation graph to drive the deformation of a mesh sequence. However, their approach does not directly address how to ensure the temporal coherence during its propagation procedure.

Constraint-based motion editing: Specifying constraints for character animation editing has been intensively explored in the past decades (refer to the latest survey by Gleicher [Gle01a]). Early work [WK88, Coh92] introduced space-time constraints into character animation editing and control. In their approaches, physical laws are employed to solve for the edited animations. Gleicher *et al.* [Gle97, Gle01b] further extended space-time constraints for character motion editing by maximally preserving critical, user-defined qualities of original motion signals. Lee and Shin [LS99] proposed a hierarchical motion curve fitting technique that adopts a multilevel B-spline representation with user-defined constraints to character animation.

In parallel to our work, recently Ikemoto *et al.* [LAF09] propose a character motion editing approach that learns a Gaussian Process model from a few given character motion editing examples, with the assistance of a full-body inverse kinematics system. Their approach shares certain similarities with our work. However, from the methodology perspective, our work extend the traditional Gaussian Process model that is used in their work to a constraint-based Gaussian Process framework, by introducing novel soft constraint-based optimization and a multi-level facial constraint model. Furthermore, there are significant differences between facial and character animation editing. For example, facial animation does not have an explicit and standard control skeleton. Also, it is often required to deal with the control of speech/emotion in facial animation editing (in our work, we use a region-based ICA scheme to deal with this issue), which is not the case for character animation editing. Therefore, we argue the above approaches cannot be directly used for facial animation editing without considerable efforts.

Motion style learning and transformation: Researchers learn statistical models including Hidden Markov Models (HMMs) [BH00], Scaled Gaussian Process Latent Variable Model (SGPLVM) [GMHP04], and behavior-specific low-dimensional spaces [SHP04] to model different styles of human motions. Liu *et al.* [LHP05] learn a physics-based character motion style from motion capture data using a novel nonlinear inverse optimization algorithm. Recently, Silva *et al.* [dSAP08] presented an interactive simulation system for stylized human locomotion. In their approach, a delicately designed controller is developed to reproduce the style of any given reference motion. In addition, the bilinear model [TF00] and the multilinear model have been successfully adapted for the transformation and transferring of facial expressions [CDB02, VBPP05].

The distinction between the above motion style learning approaches and our work is: these approaches typically focus on learning the style of existing animations, and its purpose is either for synthesizing novel motions or for transforming the style of existing motions; however, our work is aimed to learn the style of how animators *edit* existing facial animations and thus reduce the tedious manual efforts involved in current facial animation editing practices.

3. Our Approach

Fig. 1 illustrates the pipeline of our approach. It consists of the following five main steps: (1) *single/key frame editing*: given an input facial animation sequence, users can se-

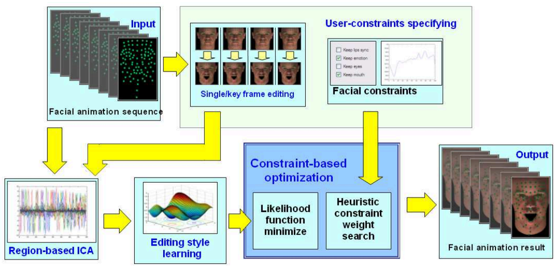

Figure 1: *The pipeline of our approach consists of the following five main steps: single/key frame editing, region-based ICA, user-constraints specifying, editing style learning, and constraint-based optimization.*

lect and edit an arbitrary number of frames via a picking-and-dragging user interface, and these selected frames are also called as "keyframes" in this paper, (2) *region-based Independent Component Analysis (ICA)*: we transform the original facial motion to a region-based ICA representation for effective model learning and intuitive user controls, (3) *user-constraints specifying*: in this step, users are allowed to impose various space-time constraints on the edited facial animation, (4) *editing-style learning*: we learn the editing style from existing facial-editing pairs (selected frames before/after the editing) using a novel constraint-based Gaussian Processes model, and (5) *soft constraint-based optimization*: Finally, other unedited frames of the animation sequence or other animation sequences are automatically "edited" by optimizing the trained Gaussian Process likelihood function with user-specified constraints. The details of the above steps will be described in follow-up sections.

3.1. Single Frame Editing

Without the loss of generality, the facial motion data for editing used in this work consist of 90 facial markers. Thus, for any facial animation frame, users can select and drag any of the 90 markers (handles) on a 3D facial mesh, and then the deformation of the whole facial geometry is computed using the linear thin-shell model [BLB*08]. In our approach, users are required to manually edit a small number of arbitrarily selected frames, and these frames do not need to be evenly distributed in the sequence. For each of the edited facial frames, its before-editing and after-editing configurations form an *facial-editing pair*. These facial-editing pairs will be used as the training set for follow-up statistical model learning.

3.2. Region-based Independent Component Analysis

Before we can learn a statistical model from the given facial-editing pairs, we need to process the original facial motion representation due to the following reasons. First, the raw facial motion (*i.e.*, the concatenation of all the 90 facial markers) is high dimensional, and thus we need to reduce its dimensionality as the first step. Second, in order to provide users high-level, intuitive user constraints/controls such as retaining the speech content while exaggerating the expression, we need to transform the facial motion into a meaningful, localized, and controllable representation. In this work, we employ Principal Component Analysis (PCA) for dimension reduction and a follow-up region-based Independent Component Analysis (ICA) to transform it to a suitable motion representation.

PCA has been widely used for data dimensionality reduction. However, in this work if PCA is directly applied to the facial motion vectors concatenating all the 90 facial markers, these is no explicit and intuitive correspondences between global PCA eigen-vectors and localized facial movements. As such, it is difficult to construct a proper mapping between high-level user constraints and facial marker configurations. To tackle this issue, we first segment the whole face (*i.e.*, the 90 facial markers) to six disjoint regions using the physically-motivated segmentation scheme [JTDP03]. These six regions are forehead, eye, the left cheek, the right cheek, mouth, and nose (Fig. 2). Subsequently, we apply PCA to the marker motion in each facial region, respectively, and retain more than 95% of its variation. The number of the retained eigen-vectors is 4 for the forehead, 8 for the eye region, 3 for the left/right cheek, 6 for the mouth region, and 3 for the nose region.

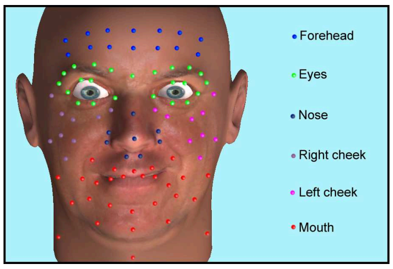

Figure 2: *Illustration of the physically-motivated face segmentation result.*

As demonstrated by Cao *et al.* [CFP03], ICA is effective for decomposing facial motions into meaningful and independent components such as expression or speech-related components. Inspired by this work, we perform ICA on region-based PCA coefficients to derive their independent components. For example, after we apply ICA to the obtained six-dimensional PCA coefficients of the mouth region, we found that two of them are more expression-correlated, while three of them are speech-correlated, based on the measuring method proposed by Cao *et al.* [CFP03]. For the other facial regions, we perform the similar ICA decomposition. Finally, we concatenate the extracted ICA coefficients of all the six facial regions of a specific frame into

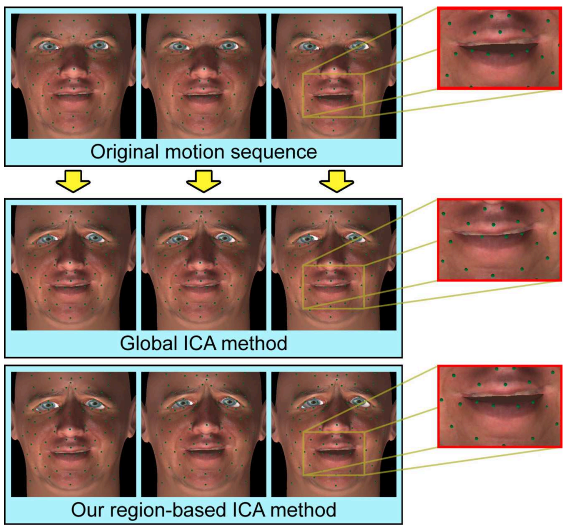

Figure 3: *An comparative editing example between the global ICA-based editing method and our region-based ICA approach. Here users want to edit the expression of the eye region from neutral to sadness while keeping its speech (lip-sync) component intact.*

a feature vector (called the *region-based ICA feature vector* of this frame). It should be noted that for each of the training editing-pairs, we will generate two such feature vectors (one for the facial pose before the editing, the other for its facial pose after the editing).

Comparing with the original global ICA [CFP03], our region-based ICA scheme provides fine-scale, localized controls. In many facial editing scenarios, users may want to edit a certain local facial region without affecting other regions. Fig. 3 shows such an example. In this example, users want to edit the expression of the eye region from neutral to sadness while keeping its speech (lip-sync) component intact. The global ICA method (the middle row of Fig. 3) fails to achieve this goal, and our region-based ICA scheme can perform this editing task with ease.

3.3. Multi-Level Facial Constraint Model

Human skeleton poses and the root trajectory as user-defined constraints have been widely recognized and used in character animation applications. However, specifying proper constraints for facial animation editing is arguably an open question. Ruttkay [Rut99] roughly classifies various facial animation constraints to the following three categories: (1) facial anatomical characteristics such as individual muscle activation, (2) behavioral repertoire such as human emotional motions, and (3) animation storyboard such as lip-sync constraints. Inspired by the Ruttkay's work, we propose a multi-level facial constraint model that combines high-level controls (*e.g.*, expression/speech), mid-level constraints (*e.g.*, mouth width and eye openness), and low-level controls (*e.g.*, individual marker constraints) for facial animation editing. As such, our multi-level facial constraint model offers sufficient flexibility and intuitiveness to users.

Our high-level facial animation constraints include the following four types: (1) emotion constraint: emotional facial motions can be retained during editing, (2) speech (lip-sync) constraint: lip-sync can be kept during editing, (3) timeline constraint: users can constrain the editing within a pre-specified timeline range, and (4) affected region constraint: users can select certain facial regions for editing, while the other regions will be kept intact. The mid-level constraints deal with facial characteristics including mouth width, mouth height, eye openness, *etc.* The low-level constraints in this work are defined as individual facial marker trajectories, *e.g.*, users can specify and edit the trajectory of any facial marker. How to incorporate these multi-level facial animation constraints into our statistical learning model will be detailed in follow-up Sections.

3.4. Facial Editing Style Learning

Now we describe how to learn a Gaussian Process model from a training set of facial editing-pairs. Gaussian Process model has been extensively used to model functional mappings between two given datasets [OK78, WR96]. In this work, the training dataset is $\langle S_i, T_i \rangle$ $(1 \leq i \leq N$, a total of N facial frames are selected for manual editing), here S_i represents the region-based ICA feature vector of the i^{th} selected, before-editing frame, and T_i represents the region-based ICA feature vector of the i^{th} selected, after-editing frame, we want to learn a Gaussian Process model, F, that is able to predict its after-editing result, y, given any unedited facial frame, x. In other words, $F(x) = y$, and both x and y represent corresponding region-based ICA feature vectors.

The Gaussian Process (GP) model offers compelling advantages over other regression methods for our specific application. First, the Gaussian Process model is non-parametric, so it does not require as much training data as many parametric regression models. Second, Gaussian Process model is context-dependent. Hence, it is able to automatically handle frames with different properties (e.g., talking or non-talking frames) in different ways. Examples are shown in Section 4.

Mathematically, a Gaussian Process model is characterized by its hyper-parameter vector θ that includes a characteristic length-scale parameter θ_1 and a signal magnitude parameter θ_2. In this work, we learn the hyper-parameter vector θ by optimizing the following marginal log-likelihood function (Eq. 1).

$$\begin{aligned} L_{GP} &= -\log\ P(T|S,\theta) \\ &= \frac{1}{2}\log|K+\sigma^2\ I| + \frac{1}{2}T^T(K+\sigma^2\ I)^{-1}T \\ &\quad +\frac{N}{2}\log\ 2\pi \end{aligned} \tag{1}$$

Here L_{GP} is the negative log-posterior of the model, σ^2 is the variance of noise (0.01 in this work), and K is a used kernel function (we use the ARD covariance function [WR96] as described in Eq. 2).

$$K_{i,j} = k(S_i, S_j) = \theta_2 \exp(-\frac{1}{2\theta_1^2}||S_i - S_j||^2) \tag{2}$$

Then, the Rasmussen's minimization algorithm [Ras06] is chosen to optimize L_{GP} due to its efficiency. The maximum number of iterations is experimentally set to 100. After θ is optimally solved, the trained Gaussian Process model yields a likelihood function for any predicted output.

3.5. Soft Constraint-based Optimization

After the Gaussian Process model is learned, we further optimize its objective function based on user-specified constraints. In this work, we treat these user-specified constraints as "soft constraints" and solve it using a constraint-based optimization algorithm.

3.5.1. Minimization of Likelihood Function

After the above Gaussian Process model is learned, for any new frame (input), x, we can obtain a distribution of its predicted output y. In addition, we can evaluate the negative log probability of the output. This log-likelihood function is shown in Eq. 3.

$$\begin{aligned} L_S &= -\log\ P(y|x,\theta) \\ &= \frac{1}{2}\log(2\pi(V(x)+\sigma^2)) + \frac{||y-U(x)||^2}{2(V(x)+\sigma^2)} \end{aligned} \tag{3}$$

$$\begin{aligned} U(x) &= \kappa(x)^T(K+\sigma^2 I)^{-1}T \\ V(x) &= k(x,x) - \kappa(x)^T(K+\sigma^2 I)^{-1}\kappa(x)^T \end{aligned}$$

Here $\kappa(x)$ is a vector in which the i^{th} entry is $k(x,S_i)$, function U returns the mean of the posterior distribution of the learned model given new input x, and function V returns the variance of the learned posterior distribution. If no constraint is involved, we just need to minimize L_S to obtain the predicted output y. However, since our approach allows users to specify multi-level facial animation constraints, the above optimization equation need to be expanded as follows.

$$minimize\ L_S(y), subject\ to\ C(y) = 0 \tag{4}$$

Here $C(y) = 0$ represents all the user-specified constraints. If all the constraints in the above optimization statement are considered as hard constraints, then typically it is difficult to solve this optimization equation due to the following reason: multi-level, user-specified constraints may conflict each other, and an ideal solution that exactly satisfies all these hard constraints may not even exist. Therefore, in this work, we treat all the user-specified constraints as "soft constraints", and the importance of each constraint can be regulated through its weight: the higher the weight of a constraint is, the higher priority it would get when the above optimization is solved.

3.5.2. Transformation of Facial Soft Constraints

We choose Sequential Quadratic Programming (SQP) to solve the above optimization equation (Eq. 4). Since multiple levels of constraints are supported in our approach, we need to first transform all these constraints to a uniform, implicit representation, $C(y) = 0$. For the sake of a clear explanation, we detail how to transform two examples of constraints to the implicit representation. Other constraints can be transformed analogously.

The first example is a high-level constraint of retaining the speech content during the editing. In other words, we need to keep the speech-related ICA components on the mouth region (detailed in Section 3.2). Its converted constraint function is shown as follows.

$$y^{speech} - x^{speech} = 0 \tag{5}$$

Here y^{speech} and x^{speech} represent the speech-related ICA coefficients of the mouth region corresponding to after-editing and before-editing cases, respectively.

For the mid-level and low-level constraints, we need to perform certain space transformations, because coordinates used in mid-level or low-level constraints are often expressed with respect to the facial marker space, while the input and output variables of the Gaussian Process model and thus the optimization equation are ICA coefficients. Here is a concrete example: if a mid-level constraint is to specify the mouth width (*i.e.*, the distance between the left mouth corner marker and the right mouth corner marker), we can convert it to the following equivalent representation (Eq. 6):

$$(E_m + P_m A_m y^m)_{lm} - (E_m + P_m A_m y^m)_{rm} = width$$

Here E_m is the mean of the mouth region vectors in the marker space, A_m is the corresponding ICA transformation matrix, P_m is the PCA transformation matrix on the mouth region, y^m is the ICA coefficients for the mouth region, and $width$ is the user-specified mouth width. The subscripts lm and rm indicate the facial markers at the left and right mouth corners, respectively.

After all the constraints are transformed to the implicit representation, $C(y) = 0$, we reformulate the above objective function (Eq. 3) by adding soft constraint terms as follows.

$$L = L_S + \sum_i^m w_i ||C_i(y)||^2 \tag{6}$$

Here w_i denotes the weight for the i^{th} constraint. We employ the SQP optimization solver [Gil06] to solve the minimization of this objective function (Eq. 6). It is noteworthy that the gradient vector used in the SQP solver is computed with respect to the region-based ICA space, not the facial marker space.

3.5.3. Heuristic Constraint Weight Search Algorithm

As described above, the importances of soft constraints are regulated through pre-specified weights; however, how to determine proper weights for these soft constraints is non-trivial. In this work, we adapt the heuristic weight determination algorithm proposed by Le *et al.* [Le96] to add exterior penalty function and increase weights until the constraints are satisfied or the optimization process ends. Its basic scheme is to increase the weight when the constraint error value is increased. Eq. 7 shows how the weight w_i is dynamically updated.

$$w_i = \frac{1 - e^{-P}}{e^{-P||C_i(y)||^2} - e^{-P}} \tag{7}$$

Here P is a control parameter (in this work, we experimentally set it to 50), $||C(y)||^2$ is the square of the constraint error that is treated as a penalty in this equation. Given the control parameter P, the weight w_i increases in accordance with the penalty value $||C(y)||^2$. This weight updating process is called at every optimization iteration. Note that in the above equation, if the penalty value $||C(y)||^2$ becomes zero, the constraint term will be forced to zero as well.

3.5.4. Comparison with Space-Time Constraints

We compared our soft constraint-based optimization algorithm with the classical space-time constraint technique proposed by Gleicher [Gle97]. The space-time constraint technique has been successfully used for interactive character animation editing. In order to perform this comparison, we adapted this space-time constraint method to fit into the facial animation editing context as follows. Following Gleicher's original method [Gle97], we concatenate the positions of all the 90 facial markers of an entire sequence into a long vector Y, and then $Y(k, j)$ represents the 3D position of the j^{th} marker at the k^{th} frame. As such, the objective function to be optimized is shown in the following equation (Eq. 8).

$$L = \frac{1}{2} Y^T \; H \; Y \tag{8}$$

Here H is a diagonal matrix formulated in Eq. 9.

$$h_{ii} = \sum_{k \in t} \sum_{j \in pt} \frac{\partial Y(k, j)}{\partial Y_i} \cdot \frac{\partial Y(k, j)}{\partial Y_i} \tag{9}$$

To be consistent with our constraint-based optimization formulation, constraints are specified in the facial marker space as follows:

$$||C_{MK}(Y)|| = 0 \tag{10}$$

Here $C_{MK}(Y)$ is a set of constraints specified in the facial marker space. SQPLAB was employed to optimize both the adapted Gleicher's approach [Gle97] and our approach. When SQP is used to solve the adapted Gleicher's method, a metric function for measuring the difference between constraints and the objective is defined in Eq. 11.

$$m(y) = a_1 \; L + a_2 \; ||C_{MK}(Y)||^2 \tag{11}$$

In this comparison, a_1 and a_2 were experimentally set to 0.1 and 5, respectively. Fig. 4 shows one of comparison results. In this example, users specify the same constraint (a mouth width constraint) to an input facial animation sequence (total 550 frames) using both the adapted space-time constraint method and our approach. As clearly shown in this figure, our approach was able to generate more desirable results (*i.e.*, closer to the original facial poses while satisfying user-specified constraints) than the adapted space-time constraint method, after the same mouth constraint was specified.

4. Selected Applications and Results

This approach is computationally efficient. When its interactive single-frame editing program runs at a laptop with the following configuration: Intel Pentium 2.4GHz CPU and 2GB main memory, it can achieve a real-time performance. In our current implementation, we use the CHOLMOD linear solver [Dav04] for sparse matrix solving, and the implementation of our constraint-based Gaussian Process model is adapted from two well-known machine learning toolkits: GPLVM [Law05] and SQPLAB [Gil06].

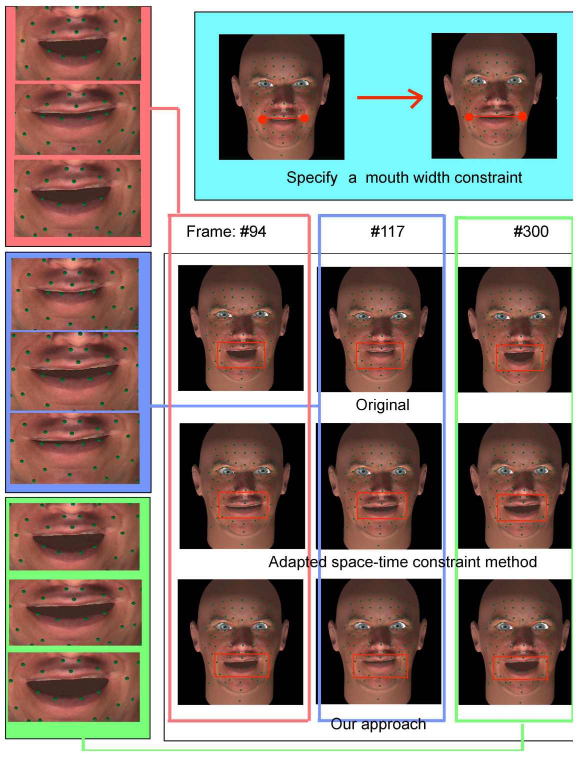
Specify a mouth width constraint
Frame: #94
#117
#300
Original
Adapted space-time constraint method
Our approach

Figure 4: *A comparison example between our approach and the adapted space-time constraint method.*

4.1. Facial Editing Style Learning

We have tested our approach on tens of facial animation sequences, and found that our approach can significantly save the manual editing efforts while producing visually appealing editing results. For example, for an input facial animation sequence consisting of 500 frames, typically our approach only need users to manually edit 20 to 30 frames to obtain desirable editing results. Another advantage of our approach is that when the number of manually edited frames is increased, the accuracy of our statistical model will be improved accordingly and thus more satisfactory editing results can be achieved.

We also compared our editing approach with the multiresolution mesh sequence editing technique proposed by Kircher and Garland [KG06]. Based on the editing of selected frames, their approach transforms motion signals into a multiresolution representation and then propagate it to the rest frames. Its whole procedure is similar to our approach; however, it should be noted that their approach allows users to manipulate any vertex of a mesh, while in our approach, user are allowed to directly manipulate only the 90 facial markers (vertices) on the mesh, and the deformations of the rest vertices are computed automatically. As such, in this comparison, the manual editing was only allowed to be performed on the 90 markers (certain vertices) on the mesh for both approaches. In addition, for both approaches, the same frames were performed with the same manual editing. Fig. 5 shows one of comparison examples. Its top panel shows three frames (#3, #25, and #179) of the original input facial animation, and these three frames were not manually edited. The bottom-left and the bottom-right panels show the results when our approach and the multiresolution editing method are applied, respectively. In this comparison, the number of manually edited frames is increased from 3 to 12, and then to 20 for both approaches. As we can see from this figure, our approach is able to generate more desirable and dynamic editing results than the multiresolution mesh sequence editing approach. Please refer to the enclosed demo video for its animation comparison results. It should be noted that when the number of the edited frames was increased, the visual quality of the resulting sequence by our approach was increased accordingly due to its adaptive learning ability; on the other hand, the results by the multiresolution approach did not benefit much from the increasing of the edited frames.

Although ground-truth comparison experiments, *e.g.*, editing a neutral speaking to an angry speaking using our approach, and then comparing its results with the ground-truth (the same person speaks the same sentence with an anger expression), would be ideal. However, it is practically infeasible to conduct these experiments due to the following major reason: Even if the same person speak the same sentences twice (*e.g.*, one is with a neutral expression, and the other is with an angry expression), and it is obvious that the phoneme sequences from these two recordings will be the same. However, the duration and phoneme timing of these two recordings (both audio and motion frames) will be significantly different. As such, after we edit the neutral recording to its intended expression (*i.e.*, anger), its total number of frames will stay the same (as the original neutral recording) and is still different from that of the original angry recording. It is noteworthy that how to perform satisfactory time-warping and alignment on two facial motion sequences with different duration is another challenging topic beyond this work. Therefore, we argue that it is extremely difficult to perform sound and fair frame-by-frame ground-truth comparisons as described above.

4.2. Editing Style Transferring Between Sequences

Our learned editing style model can be applied to not only the remaining frames of the same sequence, but also other animation sequences. This application is particularly useful when users need to edit a large number of facial animation sequences with a similar editing goal, *e.g.*, manipulating certain facial regions in a specific way or performing a similar expression transformation. Fig. 6 shows one such example. In this example, we first sculpted a sad expression from an

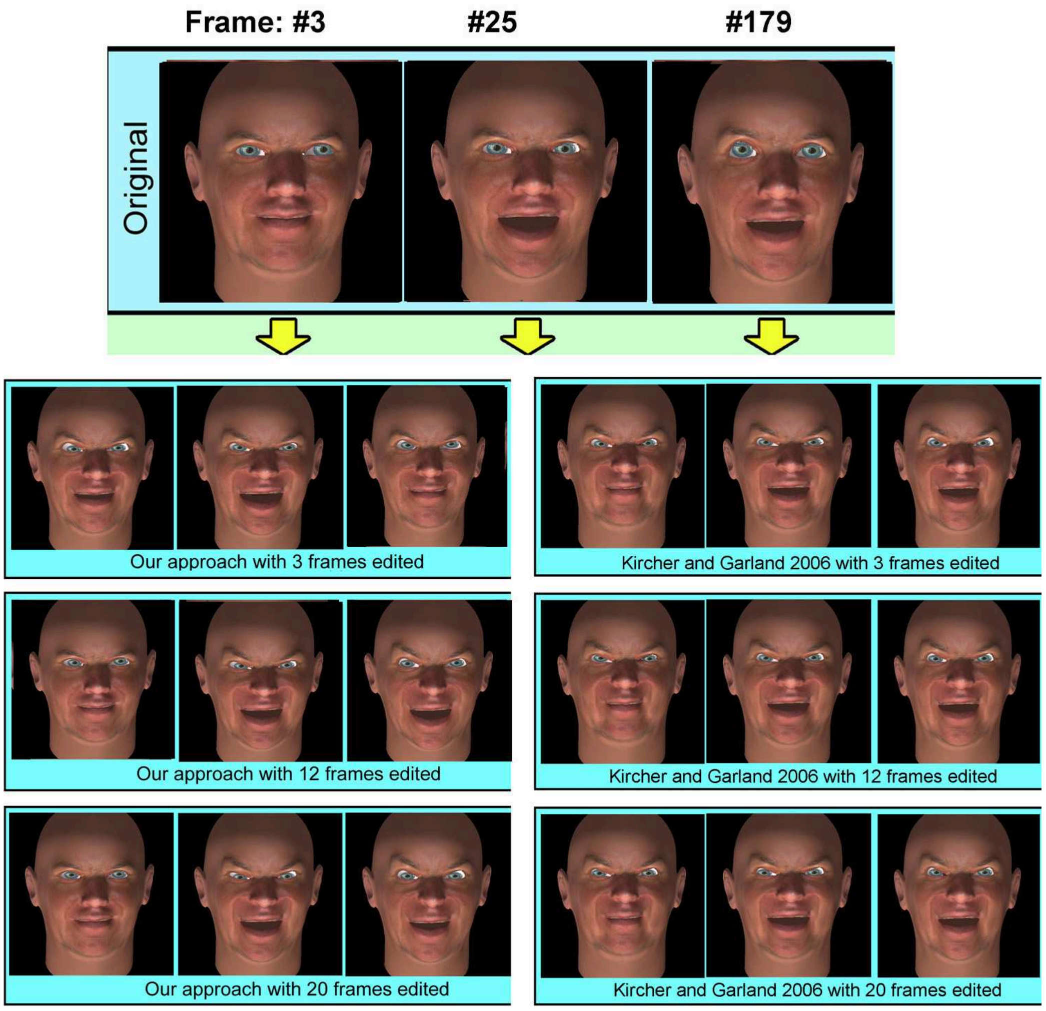
Frame: #3
#25
#179
Original
Our approach with 3 frames edited
Kircher and Garland 2006 with 3 frames edited
Our approach with 12 frames edited
Kircher and Garland 2006 with 12 frames edited
Our approach with 20 frames edited
Kircher and Garland 2006 with 20 frames edited

Figure 5: *An comparison example between our approach and the multiresolution mesh sequence editing approach. Its top panel shows three frames (#3, #25, and #179) of the original input facial animation, and these three frames were not manually edited. The bottom-left and the bottom-right panels show the results when our approach and the multiresolution editing method are applied, respectively. Note that those manually edited frames are shown in the demo video (not in this figure).*

input facial animation sequence (100 frames) by only manually editing its four selected frames (#40, #50, #83, #91, shown in the left panel of Fig. 6). Subsequently, we saved its learned editing style model and applied it onto a new facial animation sequence (100 frames) with a different avatar model. Four randomly selected frames (#1, #9, #60, and #93) of the resulting facial animation with transferred "sad-expression editing" are shown in the right panel. For the animation results, please refer to the demo video.

5. Discussion and Conclusions

In this paper, we present a novel facial editing style learning and transferring framework, and it is designed to dramatically reduce the tedious and time-consuming manual efforts involved in current facial animation editing practices. Using our approach, users are only required to edit a small number of selected frames. Besides the novel constraint-based Gaussian Process model, we also present a multi-level facial constraint model for intuitive and flexible user controls.

Certain limitations still exist in current implementation. Fig. 7 shows one example where our approach failed to produce desired results. In this example, its left panel shows the manual editing (a simple mouth pulling down operation) on a total of three keyframes. The right panel shows some of the results by our approach. The first frame in the right panel achieved the desirable result since it has the similar context

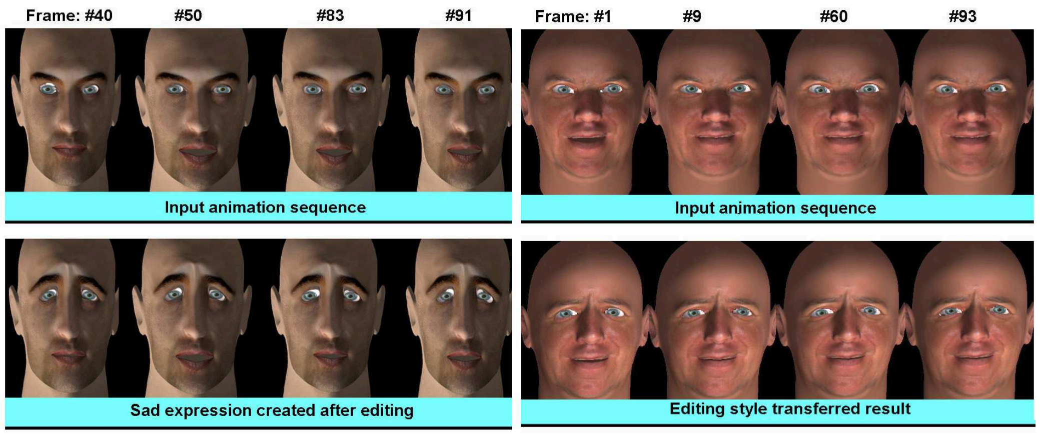

Figure 6: *The learned editing style model from an edited animation sequence (four manually edited frames are shown in the left panel) automatically transfers the learned "sad editing" to another facial animation sequence with a different avatar model (the right panel). Note that the four randomly selected frames in the right panel are not the same as those at the left panel. The face model in the left panel was acquired from http://www.turbosquid.com/.*

as one of the edited keyframes. The second and third frames in the right panel failed to keep their original eyebrow configurations since they do not have the similar context with any of the edited keyframes. Thus, one limitation of our current approach is that users have to manually identify representative keyframes for manual editing based on their own judgment, which is nontrivial at some cases. In the future, we plan to perform in-depth statistical analysis on the input animation frames, and then automatically identify and recommend those potentially critical/representative frames to users (for manual editing). Second, in certain cases, animators may want to achieve distinct editing goals or styles on different parts of an animation sequence, *e.g.*, making "happier" for the first half of the animation and "sadder" for the second half, our approach cannot guarantee the natural transition at frames of the boundary area. Finally, another limitation of our approach is that as a data-driven technique, it is hard to know in advance how many keyframes should be selected and edited in order to generate satisfactory editing results. Currently, users can conveniently edit more frames if the resulting facial animation does not sufficiently satisfy the users' expectation.

Several research issues can be further explored. First, because our current approach cannot quantify the scope of the editing styles that our model excels, it need considerable users' trial and error efforts. Exploring perceptually guided schemes [DM08] to quantify the scope of effective editing styles could be a future direction to pursue. Second, in current work, fine-scale facial details such as wrinkles are not taken into account. As such, extending or improving this framework to handle the editing of these fine-scale facial details would be useful to improve the visual realism.

Acknowledgments

This work is funded by the Texas Norman Hackerman Advanced Research Program (project number: 003652-0058-2007). Binh Huy Le is supported by a Fellowship from the Vietnam Education Foundation.

References

[BH00] BRAND M., HERTZMANN A.: Style machines. In *Proc. of SIGGRAPH '00* (2000), pp. 183–192. 2

[BLB*08] BICKEL B., LANG M., BOTSCH M., OTADUY M. A., GROSS M.: Pose-space animation and transfer of facial details. In *Proc. of SCA'08* (jul 2008), pp. 57–66. 1, 2, 3

[CDB02] CHUANG E. S., DESHPANDE H., BREGLER C.: Facial expression space learning. In *Proc. of Pacific Graphics'2002* (2002), pp. 68–76. 2

[CFP03] CAO Y., FALOUTSOS P., PIGHIN F.: Unsupervised learning for speech motion editing. In *Proc. of SCA '03* (2003), pp. 225–231. 2, 3, 4

[Coh92] COHEN M. F.: Interactive spacetime control for animation. *SIGGRAPH Comput. Graph. 26*, 2 (1992), 293–302. 2

[Dav04] DAVIS T.: Cholmod toolbox, http://www.cise.ufl.edu/research/sparse/cholmod/, 2004. 6

[DM08] DENG Z., MA X.: Perceptually guided expressive facial animation. In *Proc. Symposium on Computer Animation (SCA) '08* (Dublin, Ireland, July 2008), pp. 67–76. 9

[dSAP08] DA SILVA M., ABE Y., POPOVIĆ J.: Interactive simulation of stylized human locomotion. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers* (2008), pp. 1–10. 2

[FKY08] FENG W.-W., KIM B.-U., YU Y.: Real-time data driven deformation using kernel canonical correlation analysis. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers* (2008), pp. 1–9. 1, 2

[Gil06] GILBERT J. C.: Sqplab toolbox, http://www-rocq.inria.fr/ gilbert/modulopt/optimization-routines/sqplab/sqplab.html, 2006. 6

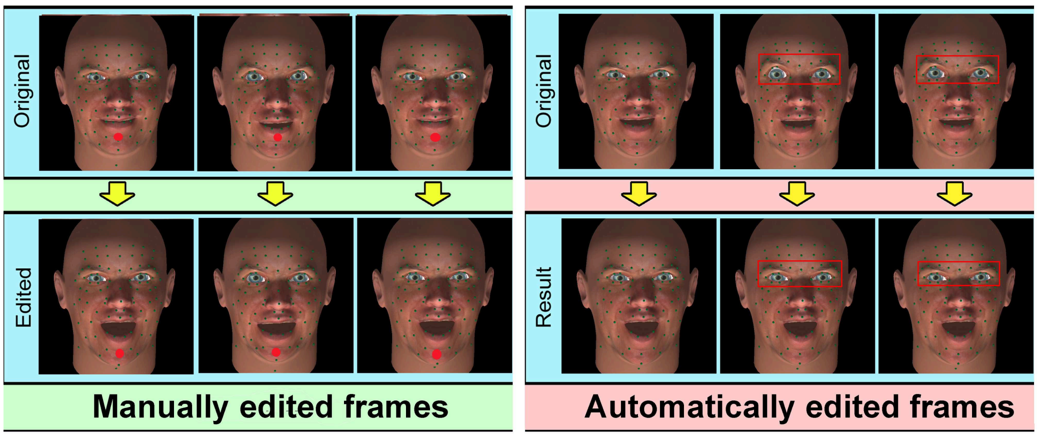

Figure 7: *An example of failed facial editing style learning and transferring by our approach due to the improper selection of representative frames (for editing).*

[Gle97] GLEICHER M.: Motion editing with spacetime constraints. In *Proc. of SI3D '97* (1997). 2, 6

[Gle01a] GLEICHER M.: Comparing constraint-based motion editing methods. *Graph. Models 63*, 2 (2001), 107–134. 2

[Gle01b] GLEICHER M.: Motion path editing. In *Proc. of I3D '01* (2001), pp. 195–202. 2

[GMHP04] GROCHOW K., MARTIN S. L., HERTZMANN A., POPOVIĆ Z.: Style-based inverse kinematics. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers* (2004), pp. 522–531. 2

[JTDP03] JOSHI P., TIEN W., DESBRUN M., PIGHIN F.: Learning controls for blend shape based realistic facial animation. In *Proc. of SCA'03* (2003), pp. 35–42. 1, 2, 3

[KG05] KIRCHER S., GARLAND M.: Progressive multiresolution meshes for deforming surfaces. In *Proc. of SCA '05* (2005), pp. 191–200. 2

[KG06] KIRCHER S., GARLAND M.: Editing arbitrarily deforming surface animations. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers* (2006), pp. 1098–1107. 2, 7

[KG08] KIRCHER S., GARLAND M.: Free-form motion processing. *ACM Trans. Graph. 27*, 2 (2008). 2

[LAF09] LESLIE I., ARIKAN O., FORSYTH D.: Generalizing motion edits with gaussian processes. *ACM Trans. Graph. 28*, 1 (2009), 1–12. 2

[Law05] LAWRENCE N.: Gaussian process toolbox, http://www.cs.man.ac.uk/ neill/gp/, 2005. 6

[LD08] LI Q., DENG Z.: Orthogonal blendshape based editing system for facial motion capture data. *IEEE Computer Graphics and Applications* (Nov/December 2008), 76–82. 2

[Le96] LE T. V.: A fuzzy evolutionary approach to constrained optimisation problems. In *Proc. of IEEE Intl Conf. on Evolutionary Computation* (1996), pp. 274–278. 6

[LHP05] LIU K. C., HERTZMANN A., POPOVIĆ Z.: Learning physics-based motion style with nonlinear inverse optimization. *ACM Trans. Graph. 24*, 3 (2005), 1071–1081. 2

[LS99] LEE J., SHIN S. Y.: A hierarchical approach to interactive motion editing for human-like figures. In *Proc. of SIGGRAPH '99* (1999), pp. 39–48. 2

[OK78] O'HAGAN A., KINGMAN J. F. C.: Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society, Series B 40*, 1 (1978), 1–42. 4

[Ras06] RASMUSSEN C. E.: Minimize function, http://www.kyb.tuebingen.mpg.de/bs/people/carl/, 2006. 5

[Rut99] RUTTKAY Z.: Constraint-based facial animation. *Constraints (Springer Netherlands)* (1999), 85–113. 4

[SHP04] SAFONOVA A., HODGINS J. K., POLLARD N. S.: Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Trans. Graph. 23*, 3 (2004), 514–521. 2

[SMND08] SUCONTPHUNT T., MO Z., NEUMANN U., DENG Z.: Interactive 3d facial expression posing through 2d portrait manipulation. In *GI'08: Proc. of Graphics Interface* (Windsor, Ontario, Canada, 2008), pp. 177–184. 2

[SSP07] SUMNER R. W., SCHMID J., PAULY M.: Embedded deformation for shape manipulation. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers* (2007), p. 80. 2

[TF00] TENENBAUM J. B., FREEMAN W. T.: Separating style and content with bilinear models. *Neural Computation 12* (2000), 1247–1283. 2

[VBPP05] VLASIC D., BRAND M., PFISTER H., POPOVIĆ J.: Face transfer with multilinear models. *ACM Trans. Graph. 24*, 3 (2005), 426–433. 2

[WK88] WITKIN A., KASS M.: Spacetime constraints. In *Proc. of SIGGRAPH '88* (1988), pp. 159–168. 2

[WR96] WILLIAMS C. K. I., RASMUSSEN C. E.: Gaussian processes for regression. In *Advances in Neural Information Processing Systems 8*. MIT Press, pp. 514–520. 4, 5

[XZY*07] XU W., ZHOU K., YU Y., TAN Q., PENG Q., GUO B.: Gradient domain editing of deforming mesh sequences. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers* (2007), p. 84. 2

[ZLGS03] ZHANG Q., LIU Z., GUO B., SHUM H.: Geometry-driven photorealistic facial expression synthesis. In *Proc. of SCA'03* (2003), pp. 177–186. 2

[ZSCS04] ZHANG L., SNAVELY N., CURLESS B., SEITZ S. M.: Spacetime faces: High-resolution capture for modeling and animation. *ACM Trans. on Graph. 23*, 3 (2004), 548–558. 1, 2