Facial Expression Editing in Video Using a Temporally-Smooth Factorization

Fei Yang[1] Lubomir Bourdev[2*] Eli Shechtman[3] Jue Wang[3] Dimitris Metaxas[1]
[1] Rutgers University, Piscataway, NJ, USA
[2] Facebook, Menlo Park, CA, USA
[3] Adobe Systems, Seattle, WA, USA
{feiyang,dnm}@cs.rutgers.edu lubomir@fb.com {elishe,juewang}@adobe.com

Abstract

We address the problem of editing facial expression in video, such as exaggerating, attenuating or replacing the expression with a different one in some parts of the video. To achieve this we develop a tensor-based 3D face geometry reconstruction method, which fits a 3D model for each video frame, with the constraint that all models have the same identity and requiring temporal continuity of pose and expression. With the identity constraint, the differences between the underlying 3D shapes capture only changes in expression and pose. We show that various expression editing tasks in video can be achieved by combining face reordering with face warping, where the warp is induced by projecting differences in 3D face shapes into the image plane. Analogously, we show how the identity can be manipulated while fixing expression and pose. Experimental results show that our method can effectively edit expressions and identity in video in a temporally-coherent way with high fidelity.

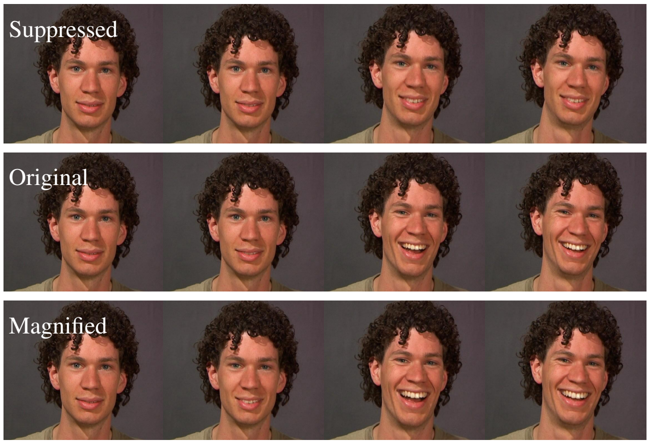

Figure 1. We can magnify or suppress an expression in video. **Middle:** Frames from the original video. **Top:** Synthesized frames in which the smile is suppressed. **Bottom:** Synthesized frames in which the smile is magnified.

1. Introduction

Video on the web is growing at astonishing rates. In 2011, on YouTube alone, every minute people upload 8 years of video content. While video capture, bandwidth and storage have become easier over time, semantic editing of video content remains a very challenging problem. Consider the problem of making a person smile in an image. This is not as simple as cutting a smile from another image, pasting it and blending the results. The entire face changes its shape, the chin becomes wider and the eyes become narrower. The appearance depends on the pose of the person as well as the identity: everyone smiles in a unique way. While a Photoshop expert with sufficient amount of time could change one's expression in an image, doing so in video is prohibitively expensive. The time dimension adds new sets of constraints, such as temporal coherence, and the temporal "signature" of an expression.

Our goal is to allow for semantic-level editing of expressions in video, such as magnifying a smile (Fig. 1) or an expression of fear, inserting an expression, or replacing unwanted expressions, such as an eye roll or facial tics. In addition, we can change the facial structure of the person, such as widen the chin or narrow the forehead, while preserving the pose and expression.

We propose a new face fitting algorithm which takes a video of a person's face and decomposes it into identity, pose and expression. This decomposition allows us to make high-level edits to the video by changing these parameters and synthesizing a new video.

We define our task as an energy minimization problem with the constraints of temporal coherence of the pose and expression and unique identity of the person in all frames. We model the face geometry over time using 3-mode tensor model, which can only deform in low-dimensional tensor

[*] This work was performed while the author was at Adobe Systems.

space. Our method results in high fidelity reconstruction and has some robustness to viewpoint variation.

2. Related Work

Manipulating and replacing facial expressions in photographs and videos has gained more attention in recent years [19]. Previous approaches fall into four categories: 3D-based, 2D expression mapping based, flow-based and image reordering based approaches.

3D-based approaches try to create photo-realistic textured 3D facial models from photographs or video, such as the expression synthesis of Pighin *et al*. [1] and the face reanimating system proposed by Blanz *et al*. [12]. Once these models are constructed, they can be used for expression interpolation. However, creating fully textured 3D models is not trivial. In order to achieve photorealism the system has to model all facial components accurately such as the eyes, teeth, ears and hair, which is computationally expensive and unstable.

2D expression mapping methods [20] extract facial features from two images with different expressions, compute the feature difference vectors and use them to guide image warping. Liu *et al*. [10] propose an expression ratio image which captures both the geometric changes and the expression details such as wrinkles. However, due to the lack of 3D information, these methods cannot deal with faces from different viewpoints. Theobald *et al*. [16] applied Active Appearance Models (AAMs) [4] to map and manipulate facial expression. Their method is based on PCA models for face appearance, and is not practical for high resolution face images.

Flow-based approaches transfer facial expression by warping face image using an expression flow map [22]. The flow map is acquired by projecting the difference between the two 3D shapes back to the 2D image plane. This method showed that accurate 3D reconstruction of the face is not necessary for transferring expressions so traditional face reconstruction methods will not help much. What is more important for generating a realistic new expression, is that the flow map should only capture typical variations of the same person, i.e., changes due to expression and not due to an identity change. Moreover, this method explicitly accommodates for small to medium changes in pose by warping the face to the correct pose before blending.

Image reordering based approaches - when the expected change in expressions is large, warping existing frames is often insufficient due to the facial appearance changes (e.g., when the mouth or eyes open). In this work we therefore combine expression flow with reordering the face frames from the entire input video using Dynamic Time Warping. A similar reordering was done to the lips region by Bregler *et al*. [2] to drive a video by audio, and by Kemelmacher *et al*. [8] to generate smooth transitions from a personal photo collection. Kemelmacher-Shlizerman *et al*. [7] demonstrated a face puppeteering method where a user is captured by a webcam and the system retrieves in real-time a similar expression from a dataset video of another person. The resulting videos are visually interesting in all of the above, however these methods were not designed for realistic expression editing in video. We use the reordering idea to swap the face region, but we keep the original pose, the face surroundings and background and we followup with an additional expression warping for a more realistic result.

Tensor factorization methods for faces - In order to separate expression from identity changes, Yang *et al*. [22] proposed a method to jointly fit a pair of face images from the same person. However, their method assumes a single dominant expression for each pair whereas our method can handle a general linear mixture of expressions and identities. We achieve that using a 3-mode tensor model that relates expression, identity and the location of the tracked feature points. A few related tensor models were introduced in the past. Vasilescu and Terzopoulos [17] proposed tensor face to model the variations in frontal face images. Their model was used for face recognition and achieved better accuracy than PCA. Vlasic *et al*. [18] built a 3D tensor model for face animation that related expressions, identity and visemes. However, these methods do not show how to directly solve the model coefficients for a new person, not in the dataset. In addition, they were not designed to work with general video sequences whereas we explicitly solve for a single identity for the entire video and require smooth variations of expression and pose for a more robust and realistic solution. Dale *et al*. [5] extended Vlasic's approach for replace facial performance in video. They could transfer expressions to a different subject that is not from the training set. However, their system requires accurate initialization of the identity parameters that relies on a commercial face reconstruction software, as well as on user interaction in one or more keyframes. To set the identity they use just the first frame, while our method is more robust to noise as we infer the identity by jointly fitting all frames of the video.

3. Joint Fitting

Our input is a video consisting of T frames of a person's face. We use a dataset of 3D face models by [18]. It consists of models of I basis identities, each in E basis expressions. The 3D geometric structure of a face is represented by a set of 3D points concatenated into a vector $s = (x_1, y_1, z_1, \cdots, x_N, y_N, z_N)^T$ that contains X, Y, Z coordinates of its N vertices. Similar to Vlasic *et al*.'s approach [18], we define a morphable face model using multilinear decomposition, which decomposes the 3D shape into

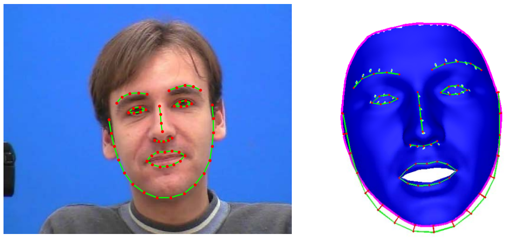

Figure 2. **Left:** Facial features detected and tracked by Active Appearance Model (AAM). **Right:** Updating face contour-landmark correspondences. The green curves connect all AAM features and the pink curve is the contour of the projected face geometry. The short red lines show the landmarks projected onto the face contour.

expression and identity:

$$s_t = \bar{s} + V \times_\beta \beta_t \times_\gamma \gamma \tag{1}$$

where s_t is the shape vector in frame t; $\bar{s}$ is the mean shape; V is core tensor of size $3N \times E \times I$; β_t is the expression coefficients in frame t. It is a vector of size E representing a linear combination of the basis expressions. γ is the identity coefficients, a vector of size I representing a linear combination of the basis identities.

Our face fitting algorithm infers the global identity γ, the expression at each frame β_t as well as the face pose in each frame, represented as a 2x3 weak perspective projection matrix. We find the values of these parameters that minimize the error between the projections of the pre-defined landmarks on the 3D face geometry, and the 2D feature points. The 2D feature points are detected and tracked by using Active Appearance Model (AAM) [4] and concatenated into a vector $Y = (x_1, y_1, \cdots, x_K, y_K)^T$. We use the face tracker proposed by Saradgih *et al.* [14] to track 66 facial features Y_t for each frame t, as illustrated in Fig. 2.

The 3D face model has a large number of vertices. We use a small portion of them which correspond to the 2D points detected by AAM. These K landmarks are predefined in 3D geometry, and can be selected by using a selection matrix $\mathbf{L}_t = L_t \otimes I_3$. The matrix L_t is a 0/1 matrix of size K by N, each row of which has exactly one entry being 1, and all others being 0. Here "$\otimes$" denotes the Kronecker product and I_3 is the identity matrix of dimension 3.

We define the projection matrix in frame t as $\mathbf{R}_t = I_K \otimes R_t$, which projects K selected vertices at the same time, where R_t is the 2x3 weak perspective projection matrix. Based on the above definitions, the 2D projections of the K selected landmarks are:

$$X_t = \mathbf{R}_t \mathbf{L}_t s_t = P_t s_t \tag{2}$$

where P_t combines our selection and projection matrices. The fitting error term E_f is defined as the sum of squared errors between the projections of the pre-defined landmarks on the 3D face geometry and the 2D feature points:

$$E_f = \sum_t ||W^{1/2}(P_t s_t - Y_t)||^2 \tag{3}$$

where $W_{2K \times 2K}$ is a positive diagonal matrix controlling the weights of landmarks. In our system we set $w = 0.5$ for eyebrow landmarks since our training shapes are textureless and these landmarks are hard to be labeled accurately. We empirically set $w = 1$ for contour points, and $w = 2$ for all other points.

In addition to minimizing the fitting error, the new shape should also be close to the distribution of the training shapes. Therefore, we define the shape energy for identity coefficients γ and expression coefficients β_t as:

$$E_\gamma = \frac{1}{2}\gamma^T \gamma \tag{4}$$

and:

$$E_\beta = \frac{1}{2}\sum_t \beta_t^T \beta_t \tag{5}$$

Finally, for a video clip, the facial expressions should change smoothly over time. Thus we also enforce temporal coherence by penalizing the 1st and 2nd order derivatives of β_t,

$$E_e = \frac{1}{2}\sum_t (\lambda_1 ||\nabla_t \beta_t||^2 + \lambda_2 ||\nabla_t^2 \beta_t||^2) \tag{6}$$

We define the total energy function as the weighted sum of the above energy terms:

$$E = E_f + \lambda_\gamma E_\gamma + \lambda_\beta E_\beta + E_e \tag{7}$$

where λ's are parameters controlling the tradeoff between the energy terms.

3.1. Optimization

The total energy E is minimized with respect to the projection matrices R_t, the expression vectors β_t and the global identity vector γ. To minimize the total energy, we use coordinate descent: in each step we optimize one variable while fixing the rest. The four steps are iterated until convergeance. Our algorithm is summarized in Algorithm 1. To initialize, we set all β_t's to zero, and γ to a random vector with unit length.

3.1.1 Fitting the projection matrices R_t

First we fit the projection matrix R_t, separately for every frame, to minimize the error between landmark projections X_t and 2D feature points Y_t. Following the restricted camera estimation method [6], which assumes that pixels are square and the skew coefficient between x and y is zero,

Algorithm 1 *Optimize E with respect to R_t, γ, β_t*

```
Initialize $\beta_t$ and $\gamma$
repeat
    Fit projection matrices $R_t$.
    Update contour-landmark correspondences $L_t$.
    Fit identity coefficients $\beta_t$.
    Fit expression coefficients $\gamma$.
until converge
```

the projection matrix R_t is parameterized with 4 unknown variables: pitch, yaw, tilt and scale. The unknown parameters can be optimized by using Levenberg-Marquardt algorithm [11] to minimize the geometric error.

3.1.2 Updating contour-landmark correspondences L_t

For landmarks located inside the face region, we can simply hard-code the corresponding 3D vertex. However, landmarks along the face contour do not have a unique corresponding vertex; they must be matched with 3D vertices along the face silhouette. In this step we build correspondences between contour landmarks and shape vertices. As shown on Fig.2, we first project the face geometry onto the image plane with projection matrix R_t. Then we find the contour of the projection (pink curve). For each landmark, we find its closest point on the contour (red lines), and assign it to the corresponding vertex.

3.1.3 Fitting the identity vector γ

The total energy E is a quadratic function of the identity coefficients γ. To minimize E with respect to γ, we set its partial derivative to zero and solve the linear system:

$$\frac{\partial E}{\partial \gamma} = \sum_t M_t^T W(P_t\bar{s} + M_t\gamma - Y_t) + \lambda_\gamma \gamma = 0 \tag{8}$$

in which:

$$M_t^{(\gamma)} = P_t(V \times_\beta \beta_t) \tag{9}$$

By solving the above equation, we get:

$$\gamma = A^{-1}B \tag{10}$$

in which:

$$A = \sum_t A_t + \lambda_1 I \tag{11}$$

where:

$$A_t = M_t^T W M_t \tag{12}$$

and:

$$B = \sum_t B_t \tag{13}$$

where:

$$B_t = M_t^T W(X_t - P_t\bar{s}) \tag{14}$$

3.1.4 Fitting the expression coefficients β_t

Since E is quadratic function of β_t, we set the partial derivative of E with respect to β_t to zero and solve the resulting linear system for β_t:

$$\begin{aligned}
\frac{\partial E}{\partial \beta_t} &= M_t^T W(P_t\bar{s} + M_t\beta_t - Y_t) + \lambda_\beta \beta_t \\
&+ \lambda_1(-\beta_{t-1} + 2\beta_t - \beta_{t+1}) \\
&+ \lambda_2(\beta_{t-2} - 4\beta_{t-1} + 6\beta_t - 4\beta_{t+1} + \beta_{t+2}) \\
&= 0
\end{aligned} \tag{15}$$

in which:

$$M_t^{(\beta)} = P_t(V \times_\gamma \gamma) \tag{16}$$

We now concatenate the β_t's in all frames into one vector $\beta = [\beta_1^T, \cdots, \beta_T^T]^T$, and solve β as:

$$\beta = A^{-1}B \tag{17}$$

in which:

$$A = diag(A_t) + \lambda_\beta I + \lambda_1 H_1 + \lambda_2 H_2 \tag{18}$$

and:

$$B = [B_1^T, \ldots, B_T^T]^T \tag{19}$$

Here A_t and B_t are defined in the same form as in Eqn. 12 and Eqn. 14, replacing M_t with $M_t^{(\beta)}$. H_1 and H_2 control the temporal smoothness and are defined as:

$$H_1 = (K_1^T K_1) \otimes I_m \tag{20}$$

$$H_2 = (K_2^T K_2) \otimes I_m \tag{21}$$

and

$$K_1 = \begin{bmatrix} 1 & -1 & \\ & \cdots & \\ & -1 & 1 \end{bmatrix}_{(T-1)\times T} \tag{22}$$

$$K_2 = \begin{bmatrix} -1 & 2 & -1 & \\ & \cdots & & \\ & -1 & 2 & -1 \end{bmatrix}_{(T-2)\times T} \tag{23}$$

3.2. Fitting a sequence

We evaluate the proposed fitting algorithm using a sequence of 46 frames from the "Talking Face Video" [3]. In this video a subject changes his expression from neutral to smile and then back to neutral. We use our fitting method to infer the expressions in each frame β_t, reduce them to 3D and visualize their trajectory over time on Fig. 3. As expected the trajectory starts from the neutral expression point, goes towards smiling and back to neutral. We plot the total energy and all the components in the first five iterations of our algorithm. The total energy monotonically decreases in each step and our method converges.

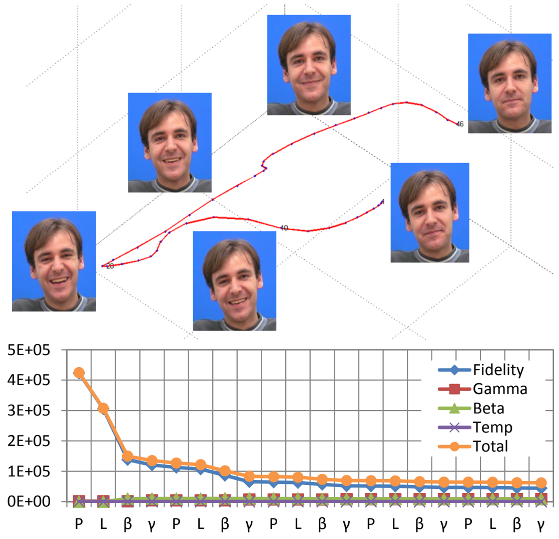

Figure 3. Fitting a sequence. **Top:** As the person goes from neutral to smiling and back to neutral expression, we are able to infer the expression coefficients over time β_t and plot their trajectory in 3D. **Bottom:** The energies after each step in the first five iterations. The total energy decreases monotonically and converges quickly.

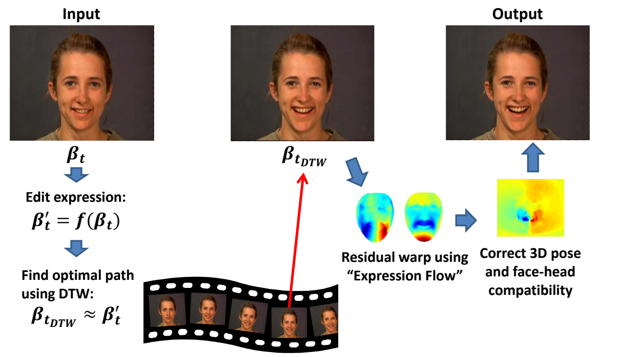
Input
Output
β_t
$\beta_{t_{DTW}}$
Edit expression:
$\beta'_t = f(\beta_t)$
Find optimal path using DTW:
$\beta_{t_{DTW}} \approx \beta'_t$
Residual warp using "Expression Flow"
Correct 3D pose and face-head compatibility

Figure 4. Expression manipulation process. For each input frame we define a desired expression manipulation $\beta'_t = f(\beta_t)$. Then we use Dynamic Time Warping to find an optimal sequence of frames from the input video $\beta_{t_{DTW}}$ that is both close to the desired expressions and is temporally coherent. We then apply "Expression Flow" to correct residual differences between and $\beta_{t_{DTW}}$ and β'_t. Finally we apply a correction warp to warp the head to match the contour of the new face (e.g. lower jaw when smiling). The entire process is automatic; the user only has to specify the location and magnitude of the expression change.

4. Expression Manipulation

Given an input video we would like to manipulate (e.g., exaggerate) the facial expressions without affecting the identity properties of the face, as well as preserve the original 3D pose of the head. Therefore we adjust the expression coefficients β_t estimated by our joint fitting algorithm according to the type of manipulation, while keeping the identity γ and pose R_t unchanged for all video frames. The adjusted coefficients β'_t could be a function of β_t or new ones (will be described in Sec. 5).

One way to obtain an image with adjusted expression from β'_t is to compute the new location of the 2D feature points and warp the input frames. The flow that warps one expression into another was called "Expression Flow" by Yang *et al*. [22][21]. However, we observed this method often does not get realistic results, especially when the change in coefficients is large. This is for two reasons: First, a facial expression (e.g. smile) contains changes in both shape and appearance (e.g. opened mouth and folds on cheeks). Only warping the shape is not enough for a realistic change in expression. Second, warping frame-by-frame requires both the source and destination to take the same time. Ideally we would like the source and destination to be able to vary in duration. Therefore we do the change in two main steps - we first apply Dynamic Time Warping (DTW) [13] to obtain a new sequence of input frames with close expression coefficients to the desired ones, and then apply "Expression Flow" to correct for the residual discrepancies. Finally we apply an addition warp to the head region to match the its boundaries to the geometry of the new expression. The flow chart of the overall manipulation process is illustrated in Fig. 4.

Dynamic Time Warping (DTW) - we treat the input video as a dataset with expressions β_t and apply the DTW method to map the sequence of new β'_t to the dataset. The distance map is computed as the Euclidean distance in the expression subspace: $D(i,j) = ||\beta'_i - \beta_j||_2$. Fig. 5 shows an example. In the original video the subject changes expression from neutral to full smile. The original expression coefficients β_t (green curve on Fig. 5, left) are scaled by a factor of 0.5 (blue curve) to neutralize the smile. Fig. 5 right shows the mapping procedure in DTW. The new sequence, with expressions $\beta_{t_{DTW}}$, only maps the first half of the original video. Therefore, the result video only shows a half smile.

Residual Expression Flow - Expression Flow $F_{ij}^{(face)}$ is a flow that warps a face with expression β_j in the original frame I_j to an expression β'_i in frame I_i and is computed as follows:

$$F_{ij}^{(face)} = R_i V \times_\beta \beta'_i \times_\gamma \gamma - R_j V \times_\beta \beta_j \times_\gamma \gamma \quad (24)$$

In our case we use Expression Flow to warp the output of DTW ($\beta_j = \beta_{t_{DTW}}$) to the desired expression ($\beta_i = \beta'_t$). Fig. 6 shows an example of exaggerating facial expression.

Correcting boundary compatibility - After applying Expression Flow to warp the face from frame $I_{t_{DTW}}$, we need to copy it into frame I_t. For a high-fidelity result, the background should also be warped, so that both sides of the

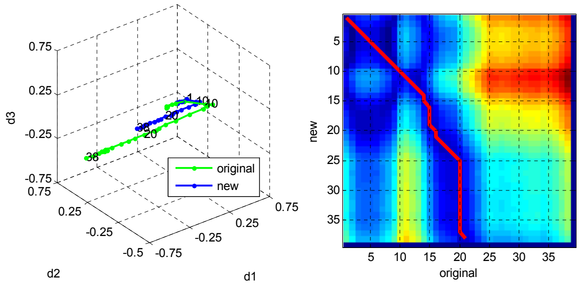

Figure 5. Expression neutralization. **Left:** The original expression coefficients β_t (green curve) is scaled by factor 0.5 (blue curve). **Right:** Frame correspondence computed using Dynamic Time Warping (red curve on top of the frame distance matrix).

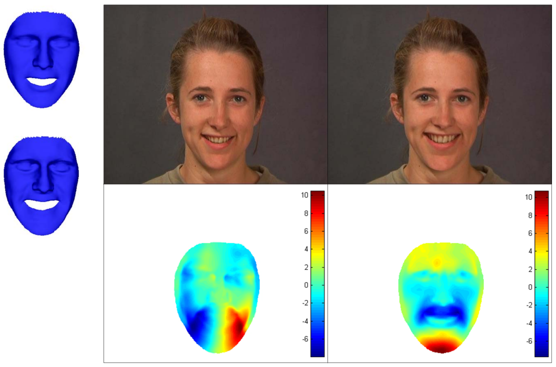

Figure 6. Exaggerating the smile using face flow. **Left margin:** The original 3D shape and, below it, the modified one after changing the expression coefficients using equation 25. **Top:** Original image (left) and the warped result (right). **Bottom:** The flow in X (left) and Y (right).

face boundary move the same way. To warp the background we compute the optical flow [9] between the two images. The face flow described above defines the warping of pixels inside the face boundary, and the optical flow defines the warping of pixels outside the face boundary. We use the Moving Least Squares [15] approach to smoothly blur the difference between the two flows.

5. Applications and Results

5.1. Changing the Magnitude of an Expression

To neutralize or exaggerate the expression, we scale the expression coefficients β_t with a factor α:

$$\beta'_t = \beta_0 + \alpha(\beta_t - \beta_0) \tag{25}$$

where β_0 are the expression coefficients of a neutral face. Setting $\alpha < 1$ will neutralize the expression, and setting $\alpha > 1$ will exaggerate the expression. The results are shown in Fig.1 and Fig.10.

5.2. Expression Interpolation and Replacement

In a similar way we can replace a section in the video (marked by the user) that contains an undesired expression. One way to do that is to interpolate linearly the expression coefficients from the two boundaries $\beta'_t = \alpha\beta_1 + (1-\alpha)\beta_2$. However this sometimes produces “frozen” looking results, especially for a long gap and similar end points. A more interesting fill can be done by letting the user choose a frame with a desired expression. Then we assign the chosen β as the value in the center of the gap and interpolate its values towards the two gap boundaries. Such an expression replacement is show in Fig.7.

5.3. Identity Modification

We can also modify the identity coefficients γ, and use the new shape to warp the original frames. An example is shown in Fig. 8. For this example, we find a subject in the training data set who has a wider chin, and use the corresponding γ_g as guidance to change γ in the input sequence as $\gamma' = \gamma + \alpha(\gamma_g - \gamma)$, where we set $\alpha = 0.5$ for this example. The result shows that by changing the identity coefficients γ, the subject’s chin widens as expected.

5.4. Limitations

While our method can produce high-fidelity face manipulation results it comes with some limitations. First, the similarity measure we use in the DTW step can capture changes related to the location of the tracked feature points. Therefore it is limited by the accuracy of the tracker and it cannot capture other subtle appearance changes (subtle lip motions, areas not covered by points such as cheeks, illumination changes). In practice we found that we can get good results as long as we copy frames from a close-by neighborhood within the same video. In the future we plan to add appearance-based features [8] to our similarity and improve the compositing method [22] to alleviate this problem. Second, when magnifying an expression, there is a limit to how much we can warp realistically a face with the residual Expression Flow, beyond the most extreme expression found in the video in the DTW step. Therefore we limit the amount of maximal warp in our implementation. Third, for some people our model does not separate well identity from expression shape changes, which causes a mixed identity and expression change when trying to edit only one of them. This is due to the linearity of our tensor model and the size of the dataset we use [18]. Lastly, our method does not perform well for large pose variations in which previously occluded part of the head would need to be synthesized.

5.5. Comparisons

We first compare our method with a single image fitting method described in [22]. This method also decomposes

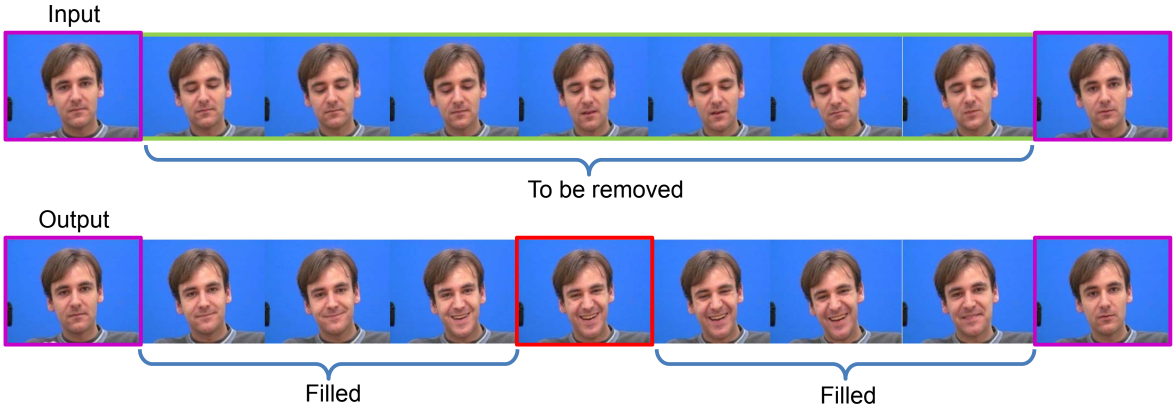

Figure 7. Replacing expression. **Top:** A section with undesired expression marked for removal. **Bottom:** The user chooses a desired frame and its expression coefficients are defined fixed for the mid frame (red). The expression coefficients are linearly interpolated in the two remaining gaps and filled using our method.

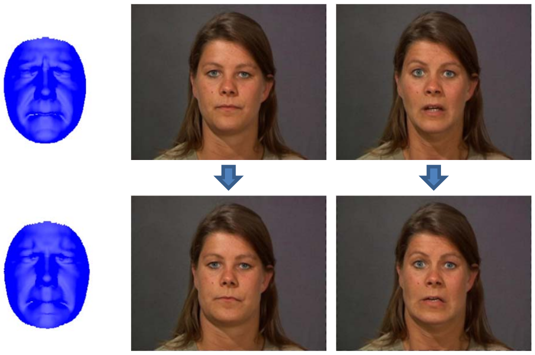

Figure 8. Changing identity coefficients γ results in changing the face structure throughout the video, independent of expression changes. **Top:** An original shape and frames. **Bottom:** After changing γ to a different person with wider chin.

the face into expression and identity coefficients. However, it operates on single images only and does not leverage temporal coherence. The method of Dale *et al.* [5] operates on the entire video. However, it fits the identity coefficients γ on the first frame only, instead of using all frames. We approximate their method by fitting γ using just the first frame. The results are shown on Fig. 9 middle. With limited number of landmarks, a single frame is not enough to find the accurate γ, which results in larger fitting error. As we show on Fig. 9 bottom, our method is able to fit the face more accurately.

6. Conclusion and Future Work

In this paper we present a new method to reconstruct 3D face shapes from a video sequence with identity constraint. By decoupling identify from expression, this method allows us to manipulate the expression in video in a variety of ways while maintaining the fidelity of the faces.

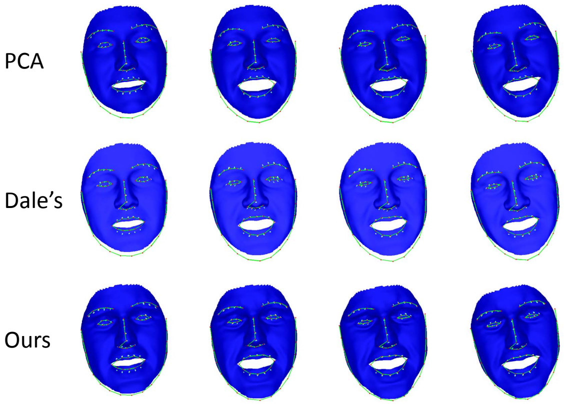

Figure 9. Comparison of the fitting result for frames 10,20,30,40. **Top:** Single image fitting of [22], which applies PCA model independently for each frame. **Middle:** An approximation of Dale *et al.* [5] using our method where we fit the identity coefficients using only the first frame. **Bottom:** Our method, which jointly fits all frames and enforces a common global identity. We are able to fit the face more accurately.

Although only used for expression manipulation, our method can be potentially used for other applications. For example, with the identity and smoothness constraints used in our optimization framework, our method is robust to tracking outliers, and could potentially be used for improving the robustness of ASM tracking in video. Our method could also support more complicated expression editing tasks, such as changing the expression statistics in a video. Finally, by combining with expression recognition techniques, our system could achieve automatic bad expression identification and replacement without any user interaction. In the future we plan to explore along these lines.

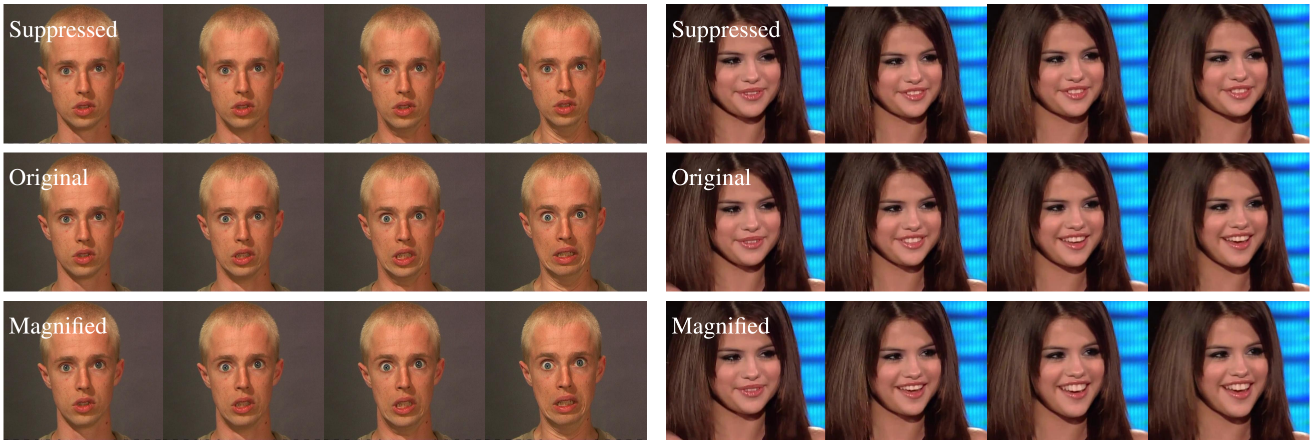

Figure 10. More examples of magnifying expressions, such as smile or fear. The original frames are in the middle rows. The synthesized expressions are shown above (suppressed) and below (magnified).

Acknowledgements

We would like to thank the anonymous reviewers for their helpful feedback. Fei Yang and Dimitris Metaxas are partially supported by the following grants to Dimitris Metaxas: NSF-0549115, NSF-0725607, NASA-NSBRI-NBTS01601, NASA-NSBRI-NBTS004, ONR-N000140910104 and Adobe Systems.

References

[1] V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating faces in images and video. *Computer Graphics Forum*, 22(3), 2003.

[2] C. Bregler, M. Covell, and M. Slaney. Video rewrite: driving visual speech with audio. In *Proceedings of ACM SIGGRAPH*, 1997.

[3] T. F. Cootes. Talking face video. http://www-prima.inrialpes.fr/FGnet/data/01-TalkingFace/talking_face.html.

[4] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[5] K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister. Video face replacement. In *Proceedings of ACM SIGGRAPH Asia*, 2011.

[6] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[7] I. Kemelmacher-Shlizerman, A. Sankar, E. Shechtman, and S. M. Seitz. Being John Malkovich. In *Proceedings of ECCV*, 2010.

[8] I. Kemelmacher-Shlizerman, E. Shechtman, R. Garg, and S. M. Seitz. Exploring photobios. *Proceedings of ACM SIGGRAPH*, 2011.

[9] C. Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. Massachusetts Institute of Technology, PhD thesis, 2009.

[10] Z. Liu, Y. Shan, and Z. Zhang. Expressive expression mapping with ratio images. In *Proceedings of ACM SIGGRAPH*, 2001.

[11] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, second edition, July 2006.

[12] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin. Synthesizing realistic facial expressions from photographs. In *Proceedings of ACM SIGGRAPH*, 1998.

[13] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoustics, Speech and Signal Processing*, 26(1):43–49, 1978.

[14] J. Saragih, S. Lucey, and J. Cohn. Deformable Model Fitting by Regularized Landmark Mean-Shift. *International Journal of Computer Vision*, 91(2):200–215, Jan 2011.

[15] S. Schaefer, T. Mcphail, and J. Warren. Image deformation using moving least squares. In *Proceedings of ACM SIGGRAPH*, 2006.

[16] B. Theobald, I. Matthews, M. Mangini, J. Spies, T. Brick, J. Cohn, and S. Boker. Mapping and manipulating facial expression. *Language and Speech*, 52:369–386, 2009.

[17] Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proceedings of the ECCV*, 2002.

[18] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. In *Proceedings of ACM SIGGRAPH*, 2005.

[19] W. Widanagamaachchi and A. Dharmaratne. 3D face reconstruction from 2D images. In *Digital Image Computing: Techniques and Applications*, pages 365–371, 2008.

[20] L. Williams. Performance-driven facial animation. In *Proceedings of ACM SIGGRAPH*, 1990.

[21] F. Yang, E. Shechtman, J. Wang, L. Bourdev, and D. Metaxas. 3D-aware appearance optimization for face morphing. In *Graphics Interface*, 2012.

[22] F. Yang, J. Wang, E. Shechtman, L. Bourdev, and D. Metaxas. Expression flow for 3D-aware face component transfer. In *Proceedings of ACM SIGGRAPH*, 2011.