Title
Automatic 3D facial expression analysis in videos

Permalink
https://escholarship.org/uc/item/3g44f7k8

Journal
Analysis and Modelling of Faces and Gestures, Proceedings, 3723

ISSN
0302-9743

Authors
Chang, Y
Vieira, M
Turk, M
et al.

Publication Date
2005

Peer reviewed

Automatic 3D Facial Expression Analysis in Videos

Ya Chang[1], Marcelo Vieira[2], Matthew Turk[1], and Luiz Velho[2]

[1] Computer Science Department, University of California
Santa Barbara, CA 93106
{yachang, mturk}@cs.ucsb.edu
[2] Instituto de Matemática Pura e Aplicada, Est. Dona Castorina, 110
Jardim Botânico, 22460-320, Rio de Janeiro, RJ, Brazil
{mbvieira, lvelho}@impa.br

Abstract. We introduce a novel framework for automatic 3D facial expression analysis in videos. Preliminary results demonstrate editing facial expression with facial expression recognition. We first build a 3D expression database to learn the expression space of a human face. The real-time 3D video data were captured by a camera/projector scanning system. From this database, we extract the geometry deformation independent of pose and illumination changes. All possible facial deformations of an individual make a nonlinear manifold embedded in a high dimensional space. To combine the manifolds of different subjects that vary significantly and are usually hard to align, we transfer the facial deformations in all training videos to one standard model. Lipschitz embedding embeds the normalized deformation of the standard model in a low dimensional generalized manifold. We learn a probabilistic expression model on the generalized manifold. To edit a facial expression of a new subject in 3D videos, the system searches over this generalized manifold for optimal replacement with the 'target' expression, which will be blended with the deformation in the previous frames to synthesize images of the new expression with the current head pose. Experimental results show that our method works effectively.

1 Introduction

Facial expression analysis and synthesis is an active and challenging research topic in computer vision, impacting important applications in areas such as human-computer interaction and data-driven animation. We introduce a novel framework for automatic facial expression editing in 3D videos. The system recognizes the expressions and replaces them by expression mapping functions smoothly. We expect to use this 3D system in the future as the core element of a facial expression analysis that takes 2D video input.

3D information is becoming widely used in this field [1-3]. A combination of image texture and 3D geometry can be used to considerably reduce the variation due to pose and illumination changes. Recent technical progress allows the capture of accurate dense 3D data in real time, which enables us to build a 3D expression database for learning the deformation space of human faces. The data capture system was developed by [4]. A coarse mesh model is fitted to track the inter-frame point motion and a dense mesh is used for synthesis of new expressions.

The nonlinear expression manifolds of different subjects share a similar structure but vary significantly in the high dimensional space. Researchers have proposed many approaches, such as locally linear embedding (LLE) [5] and Isomap [6] to embed the nonlinear manifolds in a low dimensional space. Expression manifolds from different subjects remain difficult to align in the embedded space due to various causes: (1) subjects have different face geometries; (2) facial expression styles vary by subject; (3) some persons cannot perform certain expressions; and (4) the whole expression space is large including blended expressions, so only a small portion of it can be sampled. Considering these factors, bilinear [7] and multi-linear [8] models have been successful in decomposing the static image ensembles into different sources of variation, such as identity and content. Elgammal and Lee [9] applied a decomposable generative model to separate the content and style on the manifold representing dynamic objects. It learned a unified manifold by transforming the embedded manifolds of different subjects into one. This approach assumes that the same kind of expression performed by different subjects match each other strictly. However, one kind of expression can be performed in multiple styles, such as laughter with closed mouth or with open mouth. The matching between these styles is very subjective.

To solve this problem, we built a generalized manifold that is capable of handling multiple kinds of expressions with multiple styles. We transferred the 3D deformation from the models in the training videos to a standard model. Sumner and Popovic [10] designed a special scheme for triangle meshes where the deformed target mesh is found by minimizing the transformation between the matching triangles while enforcing the connectivity. We added a temporal constraint to ensure the smooth transfer of the facial deformations in the training videos to the standard model. This model is scalable and extensible. New subjects with new expressions can be easily added in. The performance of the system will improve continuously with new data.

We built a generalized manifold from normalized motion of the standard model. Lipschitz embedding was developed to embed the manifold to a low dimensional space. A probabilistic model was learned on the generalized manifold in the embedded space as in [11].

In this framework, a complete expression sequence becomes a path on the expression manifold, emanating from a center that corresponds to the neutral expression. Each path consists of several clusters. A probabilistic model of transition between the clusters and paths is learned through training videos in the embedded space. The likelihood of one kind of facial expression is modeled as a mixture density with the clusters as mixture centers. The transition between different expressions is represented as the evolution of the posterior probability of six basic expression paths. In a video with a new subject, the deformation can be transferred to the standard model and recognized correctly.

For expression editing, the user can define any expression mapping function F: $R^6 \to R^6$, where the domain and range are the likelihood of one kind of facial expression. We currently use 3D videos as input data. Many algorithms [12,13] have been proposed to fit 3D deformable models on 2D image sequences. So the next step will be to take 2D videos as input with a system (such as [13]) used as a preprocessing module.

When the expression in the domain of F is detected, the system will search over the generalized manifold for an optimal replacement in the ‘range’ expression. The deformation of the standard model is transferred back to the subject, and blended with the facial deformation in the previous frame to ensure smooth editing. Fig. 1 illustrates the overall system structure.

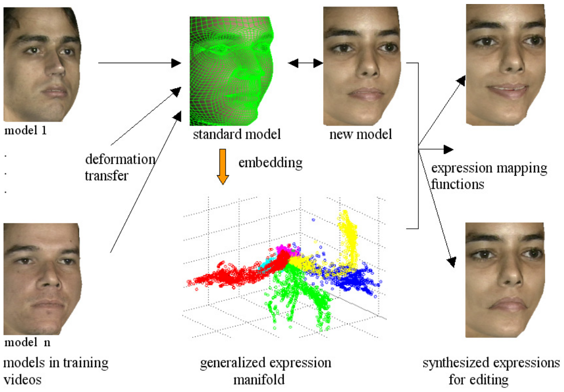

model 1
deformation transfer
standard model
new model
embedding
expression mapping functions
model n
models in training videos
generalized expression manifold
synthesized expressions for editing

Fig. 1. System diagram.

The main contributions of this paper are the following: (1) We constructed a 3D expression database with good scalability. (2) We proposed and defined a generalized manifold of facial expression. Deformation data from different subjects complement each other for a better description of the true manifold. (3) We learned a probabilistic model to automatically implement the expression mapping function.

The remainder of the paper is organized as follows. We present the related work in Section 2. We then describe how to construct the 3D expression database in Section 3. Section 4 presents how to build generalized expression manifold. Section 5 discusses the probabilistic model. Section 6 presents the experimental results. Section 7 concludes the paper with discussion.

2 Related Work

Many researchers have explored the nature of the space of facial expressions. Zhang et al. [14] used a two-layer perceptron to classify facial expressions. They found that five to seven hidden perceptrons are probably enough to represent the space of facial expressions. Chuang et al. [15] showed that the space of facial expression could be modeled with a bilinear model. Two formulations of bilinear models, asymmetric and symmetric, were fit to facial expression data.

There are several publicly available facial expression databases: Cohen-Kanade facial expression database [16] provided by CMU has 97 subjects, 481 video sequences with six kinds of basic expressions. Subjects in every video began from a neutral expression, and ended at the expression apex. FACS coding of every video is also provided. The CMU PIE database [17] includes 41,368 face images of 68 people captured under 13 poses, 43 illuminations conditions, and with 3 different expressions: neutral, smile, and blinking. The Human ID database provided by USF has 100 exemplar 3D faces. The exemplar 3D faces were put in full correspondence as explained by Blanz and Vetter [1].

Facial animation can be generated from scratch, or by reusing existing data. Noh and Neumann [18] proposed a heuristic method to transfer the facial expression from one mesh to another based on 3D geometry morphing. Lee and Shin [19] retargeted motions by using a hierarchical displacement mapping based on multilevel B-spline approximation. Zhang [20] proposed a geometry-driven photorealistic facial expression synthesis method. Example-based motion synthesis is another stream of research. Ryun et al. [21] proposed an example-based approach for expression retargeting. We improve the deformation transfer scheme in [10] by adding temporal constraints to ensure smooth transfer of source dynamics.

We were inspired by the work of Wang et al. [3]. The main difference is that we build a generalized expression manifold by deformation transfer, which is capable of handling multiple expressions with multiple styles. The probabilistic model also takes the blended expression into consideration and enables automatic expression editing.

3 3D Expression Database

To our knowledge, there is no 3D expression database publicly available, so we built our own 3D database by capturing real-time range data of people making different facial expressions. The database includes 6 subjects and 36 videos, with a total of 2581 frames. Each subject performed all six basic expressions from neutral to apex and back to neutral. The range data were registered by robust feature tracking and 3D mesh model fitting. We intend to make the database publicly available with more subjects in the near future.

3.1 Real-time 3D scanner

To construct a high quality 3D expression database, the capture system should provide high quality texture and geometry data in real-time. Quality is crucial for accurate analysis and realistic synthesis. Real-time is important for subtle facial motion capture and temporal study of facial expression.

The system used for obtaining 3D data [4] is based on a camera/projector pair and active stereo. It was built with off-the-shelf NTSC video equipment. The key of this system is the combination of the color code (*b*,*s*)-BCSL [22] with a synchronized video stream.

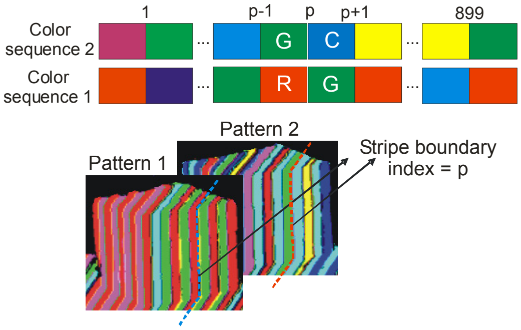


Fig. 2: Decoding stripe transitions.

The (*b*,*s*)-BCSL code provides an efficient camera/projector correspondence scheme. Parameter *b* is the number of colors and *s* is the number of patterns to be projected. Two patterns is the minimum, giving the best time coherence compromise. The complementary patterns are used to detect stripe transitions and colors robustly. Our system applies six colors that can be unambiguously detected through zero-crossings: RGBCMY. In our experiments, we use a (6,2)-BCSL code that features two patterns of 900 stripes.

To build camera/projector correspondence, we project a subsequence of these two patterns onto the scene and detect the projected stripe colors and boundaries from the image obtained by a high-speed camera. The four projected colors, two for each pattern, detected close to any boundary are uniquely decoded to the projected stripe index *p* (Fig. 2). The correspondent column in the projector space is detected in O(1) by using (6,2)-BCSL decoding process. The depth is then computed by the camera/projector intrinsic parameters and the rigid transformation between their reference systems.

We project every color stripe followed by its complementary color to facilitate the robust detection of stripe boundaries from the difference of the two resulting images. The stripe boundaries become zero-crossings in the consecutive images and can be detected with sub-pixel precision. One complete geometry reconstruction is obtained after the projection of the pattern 1 and its complement followed by pattern 2 and its complement.

The (6,2)-BCSL can be easily combined with video streams. Each 640x480 video frame in NTSC standard is composed of two interlaced 640x240 fields. Each field is exposed/captured in 1/60 sec. The camera and projector are synchronized using genlock. For projection, we generate a frame stream interleaving the two patterns that is coded with its corresponding complement as fields in a single frame. This video signal is sent to the projector and connected to the camera's genlock pin. The sum of its fields gives a texture image and the difference provides projected stripe colors and boundaries. The complete geometry and texture acquisition is illustrated in Fig. 3.

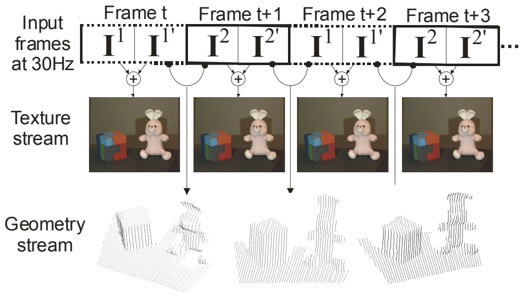
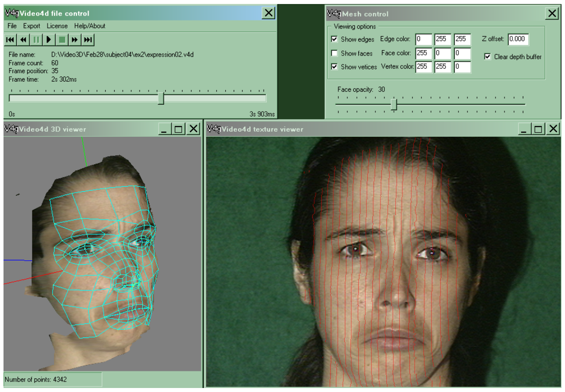


Fig. 3. Input video frames, and the texture and geometry output streams at 30 fps.

This system is suitable for facial expression capture because it maintains a good balance between texture, geometry and motion detection. Our videos were obtained by projecting 25-35 stripes over the face and the average resolutions are: vertical = 12 points/cm and horizontal = 1.25 points/cm (right bottom window of Fig. 4). We used a Sony HyperHAD camera and an Infocus LP-70 projector.

3.2 3D Data Registration



File name: D:\Video3D\Feb28\subject04\ex2\expression02.v4d
Frame count: 60
Frame position: 35
Frame time: 2s 302ms
0s
3s 903ms

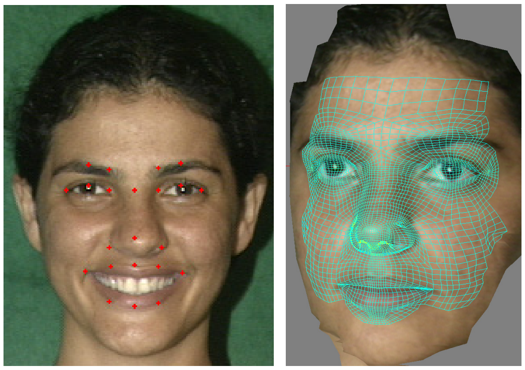
Viewing options
☑ Show edges Edge color: 0 255 255
□ Show faces Face color: 255 0 0
☑ Show vetices Vertex color: 255 255 0
Z offset: 0.000
☑ Clear depth buffer
Face opacity: 30

Number of points: 4342

Fig. 4. An example of 3D data viewer with fitted mesh

The acquired range data need to be registered for the following analysis. The range points are first smoothed by radial basis functions (RBF). We build a coarse mesh model with 268 vertices and 244 quadrangles for face tracking. A coarse generic model is fitted manually at the first frame. A robust feature tracker from Nevengineering [23] provides the 2D positions of 22 prominent feature points. The mesh's projection was warped by the 22 feature points. The depth of the vertex was recovered by minimizing the distance between the mesh and the range data [24].

An example of the 3D viewer is shown in Fig. 4. The left bottom window shows the range data with the fitted mesh. The right bottom window is the texture image with the projected 3D points. Fig. 5 (a) shows the texture image with the 22 tracked feature points. Fig. 5 (b) shows the dense mesh with 4856 vertices and 4756 quadrangles. The dense model is used for the synthesis of new expressions.



(a) (b)

Fig. 5. (a) The 2D tracking results. (b) The dense mesh model.

4 Generalized Expression Manifold

We built the generalized expression manifold by transferring the facial deformations in the training videos to a standard model. The standard model serves as the interface between the models in the training videos and models in the testing videos. The generalized manifold, that is the expression manifold of the standard model, includes all information in the training videos. The more training data we have, the better it approximates the true manifold. We can define expression similarity on this manifold and use it to search the optimal approximation for any kind of expression. The expression synthesis will involve only the standard model and target model.

4.1 Deformation Transfer with Temporal Constraints

Sumner [10] proposed a novel method to transfer the deformation of the source triangle mesh to the target one by minimizing the transformation between the matching triangles while enforcing the connectivity. This optimization problem can be rewritten in linear equations:

$$\min_{v_1 \dots v_n} \| c - Ax \|_F^2 \qquad (1)$$

where the matrix norm $\| \bullet \|_F$ is the Frobenius norm, or the square root of the sum of the square matrix elements. $v_1,\dots,v_n$ is the vertex of the unknown deformed target mesh. x is a vector of the locations of $v_1,\dots,v_n$. c is a vector containing entries from the source transformations, and A is a large sparse matrix that relates x to c, which is determined by the undeformed target mesh. This classic least-square optimization problem has closed form solution as

$$Sx = b \text{ , where } S = A'A, b = A'c\,. \tag{2}$$

The result is unique up to a global translation. We fix the rigid vertex, such as inner eyes corners to resolve the global position. x can be split as $x = [xf' \quad xm']'$ where *xf* corresponds to the fixed vertex, and *xm* to all the other vertices. Thus

$$c - Ax = c - \begin{bmatrix} Af & Am \end{bmatrix} * \begin{bmatrix} xf \\ xm \end{bmatrix} = c - Af * xf - Am * xm = d - Am * xm$$

Our goal is to transfer the deformation of a training subject in a video sequence to a standard face smoothly. The vertex v_i at frame t is represented as $v_i^t, i = 1,..., n; t = 1,..., k$. k is the length of the video. We add a constraint for temporal coherence and the optimization problem becomes

$$\min_{v_1^1 \ldots v_n^1, \ldots, v_1^k \ldots v_n^k} \sum_{t=1,\ldots,k} \| d^t - Am * xm^t \|_2^2 + \sigma \| \frac{\partial xm^t}{\partial t} \|_2^2 \tag{3}$$

where σ is the weight for temporal smoothing. c^t is the source transformation at frame t, $d^t = c^t - Af * xf$.

This problem can be solved in a progressive way by approximating

$$\frac{\partial xm^t}{\partial t} = xm^t - xm^{t-1},$$

where xm^0 is the vertex locations of the undeformed target mesh.

Eq. (3) can be rewritten as

$$\min_{v_1^1 \ldots v_n^1, \ldots, v_1^k \ldots v_n^k} \sum_{t=1,\ldots,k} \| Q * xm^t - p^t \|_2^2 \tag{4}$$

where

$$Q'Q = Am' * Am + \sigma I$$

$$Q'p^t = Am' * d^t + \sigma * xm^{t-1}$$

σ is chosen to guarantee $Am' * Am + \sigma I$ is symmetric positive matrix. Q always exists, while it is not needed to solve Q explicitly. Eq. (4) has a closed solution: $Q'Q * xm^t = Q'p^t$. For efficiency, we compute and store the LU factorization of *Q’Q* only once.

We separate the motion of the tracked source mesh into a global transformation due to head movement and a local deformation due to facial expression. The local deformation is used for facial expression (deformation) transfer.

Fig. 6 shows an example of transferring the source mesh to the target mesh with synthesized texture data.

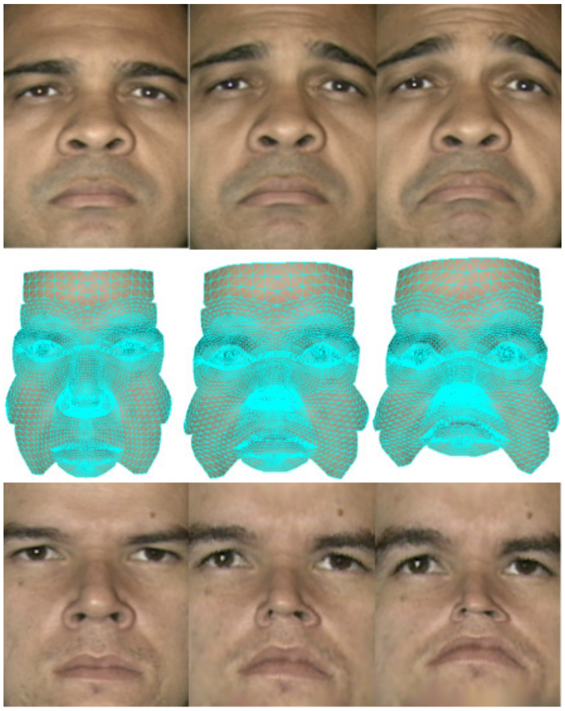


Fig. 6. Example of deformation transfer with texture synthesis. The first row is the texture image of the source video at frames 1, 12, and 24. The second row is the dense mesh of the target face with transferred deformation. The first image of the third row is the texture image of the undeformed target model. The second and the third images are the corresponding synthesized faces by the deformed dense mesh.

4.2 Lipschitz embedding

We get the deformation vectors of the standard model as $x^{s,t} \in R^{n*3}, t = 1,...k$, where n is the number of vertices; s is the number of videos and k is the length of the video. We normalize the duration of every video by re-sampling the deformation vectors at equal intervals. The interpolation is implemented by a cubic spline. We build the manifold by using the coarse mesh such that expression can be recognized quickly. The dense mesh of the standard model is saved for synthesis of the new expression.

Lipschitz embedding [25] is a powerful embedding method used widely in image clustering and image search. For a finite set of input data S, Lipschitz embedding is defined in terms of a set R of subsets of S, $R = \{A_1, A_2,..., A_k\}$. The subsets A_i are termed the reference sets of the embedding. Let $d(o; A)$ be an extension of the distance function d to a subset $A \subset S$, such that $d(o,A) = \min_{x \in A}\{d(o,x)\}$. An embedding with respect to R is defined as a mapping F such that $F(o) = (d(o; A_1); d(o; A_2);..., d(o; A_k))$.

For our experiments, we used six reference sets, each of which contains only the deformation vectors of one kind of basic facial expression at its apex. The embedded space is six dimensional. The distance function in the Lipschitz embedding should reflect the distance between points on the manifold. We use the geodesic manifold distance [5] to preserve the intrinsic geometry of the data. After we apply the Lipschitz embedding with geodesic distance to the training set, there are six basic paths in the embedded space, emanating from the center that corresponds to the neutral image. The images with blended expression lie between the basic paths.

An example of the generalized expression manifold projected on its first three dimensions can be found in the middle of the second row of Fig. 1. Points with different colors represent embedded deformation vectors of different expressions. Anger: red; Disgust: green; Fear: blue; Sad: cyan; Smile: pink; Surprise: yellow. In the embedded space, expressions can be recognized by using the probabilistic model described in the following section.

5 Probabilistic Model on the Generalized Manifold

The goal of the probabilistic model is to exploit the temporal information in video sequences in order to recognize expression correctly and find the optimal replacement for expression editing.

5.1. Model Learning

On the standard model, assume there are K videos sequences for each kind of basic expression $S, S = \{1,..., 6\}$. The embedded vector for the *i*th frame in the *j*th video for expression S is $I_{s,j,i} \in R^6$, $j = \{1,..., K\}$. By K-means clustering technique, all points are grouped into clusters $c^n, n = 1,..., r$. We compute a cluster frequency measure

$$T_{n1,n2} = \#(I_{s,j,i} \in c^{n1} \,\&\, I_{s,j,i+1} \in c^{n2}, j = 1..K, S = 1..6)$$

$T_{n1,n2}$ represents how many time the situation occurs in all videos that one frame belongs to cluster c^{n1} and its next frame belongs to cluster c^{n2}. The prior $p(c^{n2} \mid c^{n1})$ is learned as

$$p(c^{n2} \mid c^{n1}) = \begin{cases} \delta, T_{n1,n2} = 0 \\ T_{n1,n2} * scale \text{ , } otherwise \end{cases}$$

where δ is a small empirical number. Scale and δ are selected such that $\sum_{n2} p(c^{n2} \mid c^{n1}) = 1$.

The prior $p(c \mid S)$ is assigned according to the expression intensity of the cluster center, varying from 0 to 1. By Bayes' rule,

$$p(S \mid c) = \frac{p(c \mid S)p(S)}{\sum_{S} p(c \mid S)p(S)}.$$

For time series $t = 0,1,...$, the transition between different expressions can be computed as the transition between the clusters:

$$p(S_t \mid S_{t-1}) = \sum_{n1,n2} p(S_t \mid c_t = c^{n2}) p(c_t = c^{n2} \mid c_{t-1} = c^{n1}) p(c_{t-1} = c^{n1} \mid S_{t-1})$$

Due to the small variation within a cluster, S_{t-1} and S_t are conditionally independent given c_{t-1}.

5.2. Expression Recognition

Given a probe video, the facial deformation is first transferred to the standard model, and the deformation vector is embedded as $I_t, t = 0,1,...$. The expression recognition can be represented as the evolution of the posterior probability $p(S_{0:t} \mid I_{0:t})$.

We assume statistical independence between prior knowledge on the distributions $p(c_0 \mid I_0)$ and $p(S_0 \mid I_0)$. Using the overall state vector $x_t = (S_t, c_t)$, the transition probability can be computed as:

$$p(x_t \mid x_{t-1}) = p(S_t \mid S_{t-1}) p(c_t \mid c_{t-1}) \tag{5}$$

We define the likelihood computation as follows

$$p(I \mid c, S) \propto \exp[-\frac{1}{2\sigma_c^2} d(I, u_c)]\, p(c \mid S)$$

where u_c is the center of cluster c, σ_c is the variation of cluster c.

Given this model, our goal is to compute the posterior $p(S_t \mid I_{0:t})$. It is in fact a probability mass function (PMF) since S_t only takes values from 1 to 6. The marginal probability $p(S_t, c_t \mid I_{0:t})$ is also a PMF for the same reason.

Using Eq. (5), the Markov property, statistical independence, and time recursion in the model, we can derive:

$$p(S_{0:t}, c_{0:t} \mid I_{0:t}) = p(x_{0:t} \mid I_{0:t}) = p(S_0, c_0 \mid I_0) \prod_{i=1}^{t} \frac{p(I_i \mid S_i, c_i) p(S_i \mid S_{i-1}) p(c_i \mid c_{i-1})}{p(I_i \mid I_{0:i-1})}$$

By marginalizing over $c_{0:t}$ and $S_{0:t-1}$, we obtain Equation (6):

$$\begin{aligned} p(S_t \mid I_{0:t}) = \int_{c_0}\int_{S_0}...\int_{c_{t-1}}\int_{S_{t-1}}\int_{c_t} p(c_0 \mid I_0) p(S_0 \mid I_0) * \\ \prod_{i=1}^{t} \frac{p(I_i \mid S_i, c_i) p(S_i \mid S_{i-1}) p(c_i \mid c_{i-1})}{p(I_i \mid I_{0:i-1})} dc_t dS_{t-1} dc_{t-1} ... dS_0 dc_0 \end{aligned} \tag{6}$$

which can be computed by the priors and the likelihood $p(I_i \mid S_i, c_i), i = 1,..., t$. This provides us the probability distribution of the expression categories, given the sequence of embedded deformation vectors of the standard model.

5.3. Expression Editing

The user can define the any expression editing function F as needed. F: $R^6 \to R^6$.

$$F(p(S=1),...,p(S=6)) = [q_1, q_2,..., q_6]$$

where $\sum_{i=1}^{6} q_i = 1$, q is the new likelihood of one kind of facial expression. For example, if we want to edit all sadness (S=1) videos to anger (S=2), the mapping function can be defined as

$$F\ (p\ (S=1),\ p\ (S=2),\ \ldots,\ p\ (S=6))= \\ [p\ (S=2),\ p\ (S=1),\ \ldots,\ p\ (S=6)],\ \text{when}\ p\ (S=1)>\gamma. \quad (7)$$

This function will increase the likelihood of anger when the sadness is detected, that is, its likelihood is above a threshold γ.

The system automatically searches for the embedded vector with likelihood that is closest to the "range" expression. It first looks for the cluster whose center has the closest likelihood. In that cluster, the point closest to the embedded vector of the input frame is selected. We transfer the corresponding deformation vector back to the model in the new video. The deformation vector is blended with the deformation at the previous frame to ensure smooth editing. The synthesized 2D image uses the head pose in the real input frame and the texture information of the dense model.

6. Experimental Results

We collected 3D training videos from 6 subjects (3 males, 3 females). Every subject performed six kinds of basic expressions. The total number of frames in the training videos is 2581. We use Magic Morph morphing software to estimate the average of the training faces, and we use that average as the standard model. The standard model only contains geometrical data, no texture data. It will approach the "average" shape of human faces when the number of training subjects increases.

Fig. 7 includes some examples of the mesh fitting results. We change the viewpoints of 3D data to show that the fitting is very robust. A supplementary video is available at http://ilab.cs.ucsb.edu/demos/AMFG05.mpg. This video gave a snapshot of our database by displaying the texture sequences and 3D view of the range data with the fitted mesh at the same time.

Fig. 8 shows examples of deformation transfer. The motions of the training videos are well retargeted on the standard model.

Fig. 9 is an example of expression editing. The system recognized the sadness correctly and synthesized new faces with anger expression correspondingly.

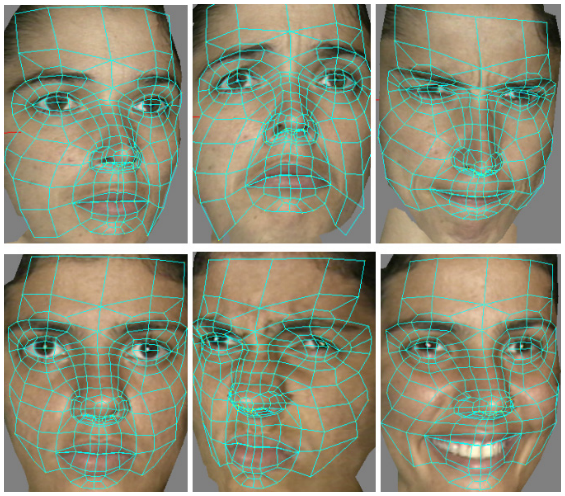


Fig. 7. Mesh fitting for training videos. Images in each row are from the same subject. The first column is the neutral expression. The second and third columns represent large deformation during the apex of expressions.

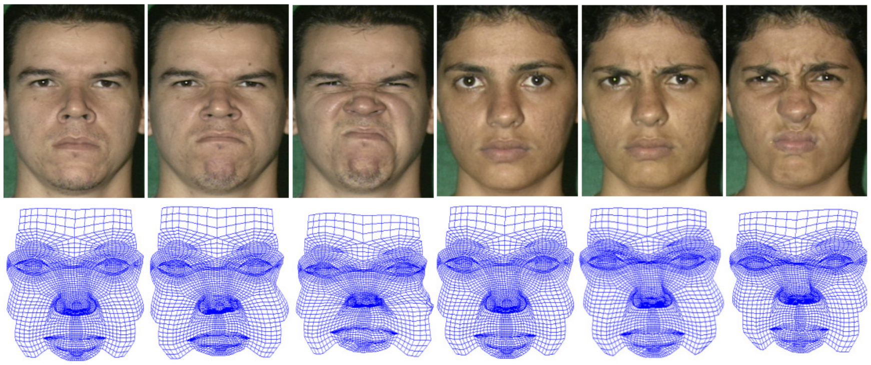


Fig. 8. Two different styles of the anger in training videos transferred to the standard mesh model. The first row and second row is images of anger and the corresponding deformed standard mesh model. The first to the third column is one style of anger at frame 1, 6, and 29. The fourth to sixth column is another style of anger at frames 1, 20, and 48.

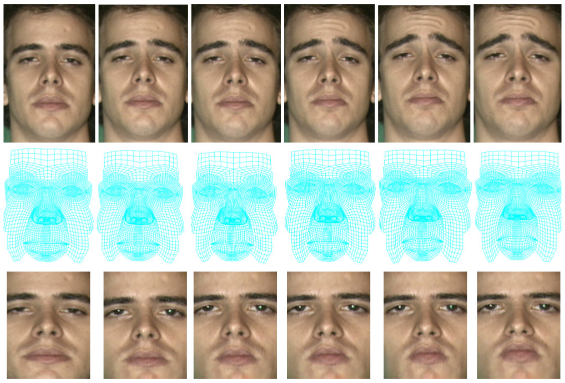


Fig. 9. Expression editing examples. First row is from the input video of sadness. We define the expression mapping function as Eq. 7. The second row is the deformed dense mesh by our algorithm. The third row is the output: the first image is unchanged, the following images are synthesized anger faces by the expression mapping function.

7 Conclusion

We introduced a novel framework for automatic facial expression analysis in 3D videos. A generalized manifold of facial expression is learned through a 3D expression database. This database provides a potential to learn the complete deformation space of human faces when more and more subjects are added in. Expression recognition and editing is accomplished automatically by using the probabilistic model on the generalized expression manifold of the standard model.

The current input is 3D videos. We plan to take 2D video input by using a system like [13]. The output video is a synthesized face with a new expression. How to separate and keep the deformation due to speech and merge the synthesized face smoothly with the background in videos [26] are important topics for the future research.

References

1. Blanz, V. and Vetter, T.: A Morphable Model for the Synthesis of 3D Face. ACM SIGGRAPH, Los Angeles, CA, (1999) 187-194.
2. Zhang, Y., Prakash, E.C., Sung, E.: A New Physical Model with Multi-layer Architecture for Facial Expression Animation Using Dynamic Adaptive Mesh. IEEE Transactions on Visualization and Computer Graphics, 10(3):339-352, (2004).
3. Wang, Y., Huang, X., Lee, C., Zhang, S., Li, Z., Samaras, D., Metaxas, D., Elgammal, A., Huang, P.: High Resolution Acquisition, Learning and Transfer of Dynamic 3-D Facial Expressions. Proc. Eurographics 2004, Grenoble, France, (2004).

4. Vieira, M.B., Velho, L., Sá, A., Carvalho, P.C.: A Camera-Projector System for Real-Time 3D Video. IEEE Int. Workshop on Projector-Camera Systems, San Diego, CA, (2005).
5. Roweis, S., Saul, L.: Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science, 290; 2323-2326, (2000).
6. Tenenbaum, J.B., Silva, V. de, Langford, J.C.: A Global Geometric Framework For Nonlinear Dimensionality Reduction. Science, vol. 290, pp. 2319-2323, (2000).
7. Tenebaum, J.B., Freeman, W.T.: Separating Style and Content with Bilinear Models. Neural Computation J., Vol. 12, pp. 1247-1283, (1999).
8. Vasilescu, A.O., Terzopoulos, D.: Multilinear Subspace Analysis for Image Ensembles. Proc. Computer Vision and Pattern Recognition, Madison, WI, (2003).
9. Elgammal, A., Lee, C.: Separating Style and Content on a Nonlinear Manifold. Proc. Computer Vision and Pattern Recognition, Washington, (2004).
10. Sumner, R., Popovic, J.: Deformation Transfer for Triangle Meshes. ACM SIGGRAPH, Los Angeles, CA, (2004).
11. Chang, Y., Hu, C., Turk, M.: Probabilistic Expression Analysis on Manifolds. Proc. Computer Vision and Pattern Recognition, Washington, (2004).
12. Vacchetti, L., Lepetit, V., Fua, P.: Stable Real-time 3D Tracking Using Online and Offline Information. IEEE Trans. on Pattern Analysis and Machine Intelligence, 26 (2004) 1385–1391.
13. Goldenstein, S.K., Vogler, C., Metaxas, D.: Statistical Cue Integration in DAG Deformable Models. IEEE Trans. on Pattern Analysis and Machine Intelligence*,* 25 (2003) 801–813.
14. Zhang, Z., Lyons, M., Schuster, M., Akamatsu, S.: Comparison Between Geometry-based and Gabor-wavelets-based Facial Expression Recognition Using Multi-layer Perceptron: IEEE Conf. on Automatic Face and Gesture Recognition, Nara, Japan, (1998).
15. Chuang, E., Deshpande, H., Bregler, C.: Facial Expression Space Learning. Pacific Graphics, (2002).
16. Kanade, T., Cohn, J., Tian, Y.: Comprehensive Database for Facial Expression Analysis. IEEE Conf. on Automatic Face and Gesture Recognition, (2000) 46-53.
17. Sim, T., Baker, S., Bsat, M.: The CMU Pose, Illumination, and Expression Database. IEEE Trans. on Pattern Analysis and Machine Intelligence, 25 (2003) 1615-1618.
18. Noh, J., Neumann, U.: Expression Cloning. ACM SIGGRAPH, Los Angeles, CA, (2001).
19. Lee, J., Shin, S.Y.: A Hierarchical Approach to Interactive Motion Editing for Human-like Figures. ACM SIGGRAPH, Los Angeles, CA, Los Angeles, CA, (1999) 39-48.
20. Zhang, Q., Liu, Z., Guo, B., Shum, H.: Geometry-Driven Photorealistic Facial Expression Synthesis. SIGGRAPH Symposium on Computer Animation, (2003).
21. Pyun, H., Kim, Y., Chae, W., Kang, H.W., Shin, S.Y.: An Example-Based Approach for Facial Expression Cloning. Siggraph Symposium on Computer Animation, (2003).
22. Sá, A., Carvalho, P.C., Velho, L.: (b,s)-BCSL: Structured Light Color Boundary Coding for 3D photography. Int. Fall Workshop on Vision, Modeling, and Visualization, (2002).
23. www.nevengineering.com
24. Sederberg, T.W., Parry, S.R.: Free-Form Deformation of Solid Geometric Models. ACM SIGGRAPH, pp. 151-159, Dallas, TX, (1986).
25. Bourgain, J.: On Lipschitz Embedding of Finite Metric Spaces in Hilbert Space. Israel J. Math., 52 (1985) 46-52.
26. Blanz, V., Scherbaum, K., Vetter, T., Seidel, H.: Exchanging Faces in Images. Proc. Eurographics, Grenoble, France, (2004).