A Data-driven Approach for Facial Expression Synthesis in Video

Kai Li[1,2] Feng Xu[1] Jue Wang[3] Qionghai Dai[1] Yebin Liu[1]
[1]Department of Automation, Tsinghua University
[2]Graduate School at Shenzhen, Tsinghua University [3]Adobe Systems

Abstract

This paper presents a method to synthesize a realistic facial animation of a target person, driven by a facial performance video of another person. Different from traditional facial animation approaches, our system takes advantage of an existing facial performance database of the target person, and generates the final video by retrieving frames from the database that have similar expressions to the input ones. To achieve this we develop an expression similarity metric for accurately measuring the expression difference between two video frames. To enforce temporal coherence, our system employs a shortest path algorithm to choose the optimal image for each frame from a set of candidate frames determined by the similarity metric. Finally, our system adopts an expression mapping method to further minimize the expression difference between the input and retrieved frames. Experimental results show that our system can generate high quality facial animation using the proposed data-driven approach.

1. Introduction

Performance-driven facial animation has been in the spotlight since the 1980s. It refers to the problem of mapping facial performance from one identity to another, with the goal of making the rendered facial animation of the target identity to be both realistic and consistent with the source performance.

Although tremendous progress has been achieved in the past few decades, the problem remains unsolved. Previous approaches have mainly focused on expression fidelity, that is, making the rendered facial expression on the target face to be perceptually close to the input expression of the input face. On the other hand, photorealistic rendering has been largely ignored, and previous approaches often use a 3D face model as the target avatar. It is unclear how to render photorealistic facial animations of a real person's face, given the performance of another person. Furthermore, many previous approaches heavily rely on additional information such as markers on the source face [1, 17], or intensive user interaction for tracking [28, 15]. The application range and efficiency of these methods are thus limited.

In this paper, we aim at developing an automatic system to transfer facial performance in a face video to another person, resulting in a natural-looking video of the target person. Inspired by the recent data-driven approaches on occluded face completion [8] and human motion animation [29], our system relies on an existing expression database of the target person to achieve this goal. Since the database provides natural video frames of the target face under various expressions, we can use them as reference to render a new video that matches with the input performance. However, this is not a trivial task, given the following main challenges:

1. how to measure the expression similarity between video frames of different identities;
2. how to effectively search the database to ensure that the generated video is not only close to the input performance, but also temporally coherent;
3. since the database is limited in size and cannot cover all expressions in the input performance, how to further adjust the expression in a target video frame to improve its expression accuracy.

Our system employs a set of techniques to address these challenges. Specifically, we propose a novel metric for measuring the similarity between expressions of different identities in video. For balancing between temporal coherence and expression matching accuracy, we first find k-nearest neighbors from the database as candidates for each input frame, and solve for the optimal output sequence using an optimization approach. Finally, to account for some subtle expression difference between each input and retrieved frame, we propose an expression transfer method and use its result to further refine the expression of the retrieved frame. Experimental results show that our system is able to synthesize temporally-coherent, photorealistic facial animation video that matches well with the input performance.

2. Related Works

This work is related to previous research on facial expression matching, facial expression retargeting (mapping),

978-1-4673-1228-8/12/$31.00 ©2012 IEEE

and video-to-video synthesis.

Facial Expression Matching. Our requirement is to find the most similar expression, instead of classifying facial motion into discrete, pre-defined classes. Features used in expression recognition community, such as Gabor Wavelets [19], LBP [21] and FACS [9], might provide an alternative approach to our goal. However, they often fail to take into account of identity difference. A smile with moustache, for example, is different from another smile without moustache in terms of LBP feature. Only with plenty of training examples with and without moustache, the classifiers may tell that the two smiles are the same. Furthermore, these metrics may not infer a continuous real-valued distance measure, which means that they are often not accurate enough to capture subtle expression difference. CERT [14], for example, only recognizes Action Units of peak expressions well. It is unclear how well it can discriminate subtle AU movement. In contrast, these two major problems do not exist in our proposed expression similarity measure.

3D Model-based Facial Performance Retargeting. There has been extensive work on facial performance modeling and retargeting. In PCA-based models, such as AAM [4]/CLM [23], 3D morphable model [3], multilinear model [25, 7], and deformable model [27], generic basis is learned from large training data by preserving the principal components. They try to track all expressions robustly at the cost of giving up the fine details. While in our refinement approach, with two images of similar expressions, the optical flow between them can better capture the fine detailed expression difference, which leads to more accurate retargeting result. Blendshape model of a specific character can be established for realtime animation [26]. However, the number of blendshapes is a contradiction between coverage of the model and suitability in tracking. Some other systems [22, 20, 2] try to create textured photorealistic 3D facial models. However, it is not easy to acquire the fully textured 3D models.

Image-based Expression Mapping. Some face synthesis systems directly operate on 2D images to achieve expression transfer. Williams's system [28] extracts facial features from the source and target images, and uses the feature difference to guide the warping. Liu *et al*. [17] propose an Expression Ratio Image (ERI) to enhance expression mapping by capturing the illumination changes. Zhang *et al*. [30] use geometry component to compute texture image of each subregion by compositing the example face images together. However, these methods usually cannot deal well with a large topology change between two images. Our method overcomes this limitation by retrieving a target face from the database which has similar expression to the input face. Besides, these methods usually are labor intensive.

Video-to-Video Synthesis. Our work is also related to previous video-to-video synthesis systems. Similar to our goals, Kemelmacher-Shlizerman *et al*. [10] utilize a database to synthesis a new face video of a target person, driven by a face video of another person. However, their system mainly focuses on measuring facial expression similarity under large pose difference. The final video is simply created by lining up the individually most similar images, which may not be temporally coherent. The video face replacement system [7] replaces the face in a target video with the face in a source video while maintaining spatial-temporal consistence. However, it assumes the coarse semantic correspondence and roughly approximate appearance between source and target videos.

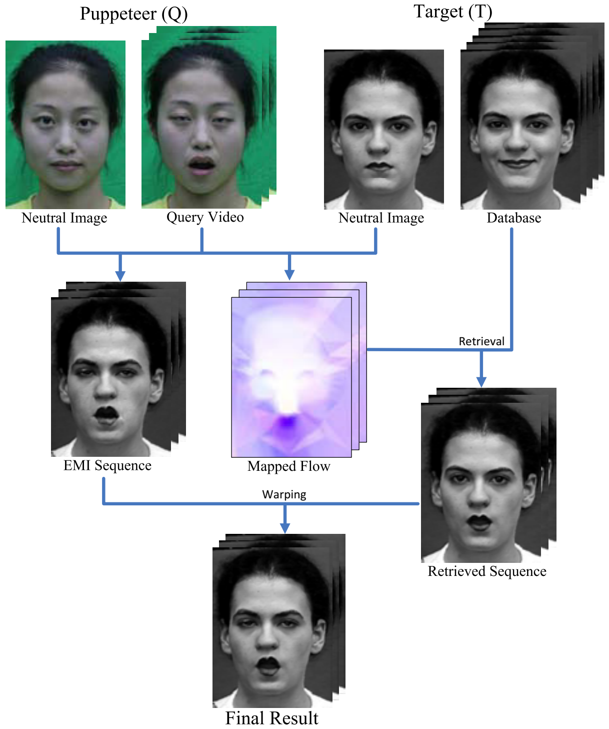


Figure 1. System overview. The optical flow between each query frame and its neutral face is first mapped to the target person, which is used to perform retrieval from the database. Meanwhile, the neutral image of the target person is warped to generate an EMI sequence with the query performance. Finally, the retrieved sequence is refined by the EMI to synthesize the final result.

3. System Overview

Figure 1 shows the overview of our system. To generate realistic expressions for a target person, we first capture a short video of this person performing some basic expressions, such as anger, fear, surprise, sadness, joy, and disgust. Given a facial performance video of another identity, whom we call the puppeteer, our method tries to synthesize the same performance using the database of the target person.

Specifically, for each input frame, we query the database and find k frames of the target person that have the most similar expressions to the input frame, using an optical flow based similarity metric described in Section 4.1. Instead of directly taking the most similar frame to form a matched sequence, as did in the system of Kemelmacher-Shlizerman *et al*. [10], we formulate the task of finding the optimal consecutive frames as a shortest path problem, as described in Section 4.2. The obtained sequence contains similar expressions to the puppeteer, and is also temporally coherent.

However, due to the limited size of the database, finding a perfect expression match for every input frame is almost impossible, not to mention that some expressions of the puppeteer may have unique characteristics. To account for the subtle expression difference between the input frame and the retrieved frame, we apply an expression mapping technique to generate another candidate face which we call the EMI image, as described in Section 4.3. The EMI image usually has more accurate facial expression than the retrieved frame, but the facial appearance in it may contain significant artifacts. In the final step, we combine the EMI image and the retrieved frame together to generate the final output frame, which has both accurate expression and realistic appearance, as described in Section 4.3.

4. The Algorithm

4.1. Expression Similarity Metric

Given a face image Q_e of the puppeteer, our system tries to find a corresponding face image T_e from the database of the target person, which has the most similar facial expression to Q_e. To achieve this we need a facial similarity metric which can accurately measure the expression difference between Q_e and T_e, while ignoring the identity/appearance difference between the two images.

To develop such a metric, our system uses the neutral faces of both the puppeteer and the target person, denoted as Q_n and T_n, respectively. T_n only needs to be identified once when we build the database, and we assume Q_n is marked by the user in the input video. To describe how the face of the puppeteer changes from Q_n to Q_e, we can compute an optical flow field [16] between these two images, denoted as $\boldsymbol{F}_{Q_n \to Q_e} \in \mathbb{R}^{m \times 2}$, where m denotes all face pixels in Q_n. To remove the global head motion from the flow field, we estimate a 2D similarity transformation using the nose region which is mostly invariant to expression change, and align Q_e to the pose of Q_n before computing the expression difference. We also normalize the flow using the width of the face. Similarly, a flow field $\boldsymbol{F}_{T_n \to T_e} \in \mathbb{R}^{n \times 2}$, where $n \neq m$, can be computed between T_n and T_e. However, we cannot directly compare these two flow fields due to the identity/appearance difference.

To establish an accurate correspondence between the two flow fields, we first detect facial landmarks only on neutral faces Q_n and T_n, using the standard Active Shape Model (ASM) [5], which works particularly well on frontal faces with neutral expression. However it does not cover the entire face as needed in the later steps of our algorithm. We thus manually label more landmark points on the two neutral faces, as shown in Figure 2(a). We then triangulate the face region in Q_n and T_n using Delauney triangulation, which leads to a pixel-wise registration function $g : Q_n \to T_n$. Furthermore, since the semantic correspondence between two identities should be invariant to different facial expressions, it is reasonable to assume that $g' : Q_e \to T_e$, the registration function between two expression images, approximately equals to $g : Q_n \to T_n$. Given the registration function, for a point $\vec{a} \in Q_n$ which moves to $\vec{a}' \in Q_e$, its corresponding flow vector on T_n is computed as:

$$\Delta \vec{b} = g(\vec{a}') - g(\vec{a}), \tag{1}$$

where $\vec{b} = g(\vec{a})$ is the corresponding point of $\vec{a}$ on T_n.

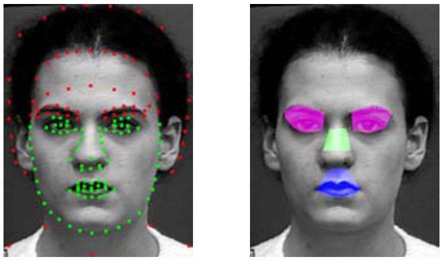

(a) facial markers (b) local regions

Figure 2. Initialization of the neutral face. (a) Green markers are detected by ASM, and red ones are manually labeled. (b) The eye, mouth and nose regions are marked in magenta, blue, and green, respectively.

By applying this mapping for all face pixels in Q_n, we obtain a mapped flow field $\boldsymbol{F}'_{Q_n \to Q_e}$, which now can be compared with $\boldsymbol{F}_{T_n \to T_e}$ to measure how similar the two expressions are. As pointed out in previous work [10], the main source of expression difference comes from the eye and mouth regions, we thus only use pixels inside these regions to compute the expression similarity (see Figure 2(b)). A straightforward approach is to measure the expression difference between Q_e and T_e by looking at the absolute flow difference:

$$\begin{aligned} d_e(Q_e, T_e) = \alpha_e \sum_{i \in \text{eye}} \left\| F'_{Q_n \to Q_e, i} - F_{T_n \to T_e, i} \right\| \\ + \alpha_m \sum_{i \in \text{mouth}} \left\| F'_{Q_n \to Q_e, i} - F_{T_n \to T_e, i} \right\|, \end{aligned} \tag{2}$$

where the subscription i refers to the ith row in the flow matrix $\boldsymbol{F}$. $\alpha_{\{e,m\}}$ are the weights for the eye and mouth regions, respectively.

The distance metric in Equation 2 works well in most cases in our experiments, however we found that it occasionally makes mistakes as shown in Figure 3. This is be-

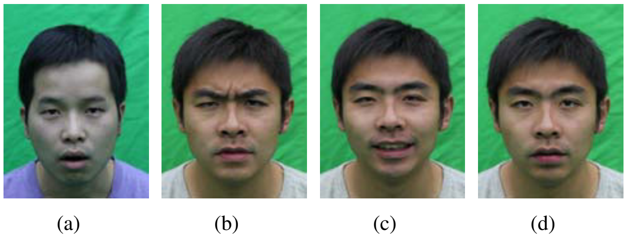


Figure 3. Comparison of expression metrics. (a) Query frame. (b) The most similar expression chosen by LBP method, which has an unwanted frown. (c) The most similar expression chosen by the metric in Equation 2, which has a smile rather than surprise. (d) The most similar expression chosen by the metric in Equation 4, which matches well with the query frame.

cause in Equation 2, we only consider the magnitude of the flow difference $\boldsymbol{F}'_{Q_n\to Q_e} - \boldsymbol{F}_{T_n\to T_e}$. However, flow direction usually contains more critical information about facial expression. For example, smile is always associated with upward motion of the mouth corners, while cry is usually accompanied with downward motion of them. This suggests that the expression in Q_e is significantly different from that in T_e if $\boldsymbol{F}'_{Q_n\to Q_e}$ has a very different direction from $\boldsymbol{F}_{T_n\to T_e}$, even when the magnitude of the difference is small. With this observation, we re-design the expression distance metric as:

$$d_b(\vec{u},\vec{v}) = \beta_m|\vec{u}-\vec{v}| + \beta_o(-\vec{u}\cdot\vec{v} + |\vec{u}||\vec{v}|), \tag{3}$$

$$\begin{aligned} d_e(Q_e,T_e) = {} & \alpha_e \sum_{i\in\text{eye}} d_b(F'_{Q_n\to Q_e,i}, F_{T_n\to T_e,i}) \\ & + \alpha_m \sum_{i\in\text{mouth}} d_b(F'_{Q_n\to Q_e,i}, F_{T_n\to T_e,i}), \end{aligned} \tag{4}$$

where $\beta_{\{m,o\}} \in [0,1]$ are the weights for the magnitude and orientation terms, respectively, and subjected to $\beta_m + \beta_o = 1$. When β_o equals to zero, Equation 4 reduces to Equation 2. The offset $|\vec{u}||\vec{v}|$ in Equation 3 makes sure that the orientation term is nonnegative. Note that this distance metric does not preserve symmetry and triangle inequality. To make it more mathematically sound, one could compute the inverse distance $d_e(T_e, Q_e)$ and use the average of these two as the final distance metric. However in practice we do not find it to be necessary, as $d_e(Q_e, T_e)$ can well describe the facial expression difference between two images with different identities, and is sufficient for our application.

4.2. Retrieval-based Video Synthesis

Using the similarity metric defined above, a naive approach for video synthesis is to for each input frame, find its nearest neighbor in terms of expression in the database, and stack them together to form the final output video. However, we found this approach does not work well, as the temporal coherence of facial expression in the final video is not well maintained, and the final video often appears to be jittering. The supplementary material contains videos that illustrate this problem. Our system employs some additional techniques to solve the temporal coherence problem, as we will describe in detail in this subsection.

4.2.1 Incorporating Expression Velocity

First, the distance metric defined in Equation 4 only concerns the expression similarity between two faces. However in video, we also need to care about the velocity of expression change at each frame. The most similar frame should be the one whose expression and expression velocity both agree with that of the query frame. To measure expression velocity, we simply compute another optical flow between the current and the next frame in a video sequence. Let $Q_e^{(q)}$ be the qth query frame, its expression velocity is computed as:

$$\mathrm{d}\boldsymbol{F}_{Q_e^{(q)}} = \boldsymbol{F}_{Q_e^{(q)}\to Q_e^{(q+1)}}. \tag{5}$$

Similarly, for a frame $T_e^{(t)}$ in the database, we calculate its expression velocity as $\mathrm{d}\boldsymbol{F}_{T_e^{(t)}}$. Again, due to the identity and expression difference between $Q_e^{(q)}$ and $T_e^{(t)}$, it is unwise to directly compute the distance between $\mathrm{d}\boldsymbol{F}_{Q_e^{(q)}}$ and $\mathrm{d}\boldsymbol{F}_{T_e^{(t)}}$. It is necessary to warp the expression velocity flow fields to remove the identity difference, as we did in Section 4.1. We also need to warp both velocity flow fields to map them onto the neutral expression to remove the expression difference between them.

Specifically, for the database frame, we apply the inverse optical flow of $\boldsymbol{F}_{T_n\to T_e^{(t)}}$ on $\mathrm{d}\boldsymbol{F}_{T_e^{(t)}}$, resulting in the warped expression velocity flow $\mathrm{d}\boldsymbol{F}'_{T_e^{(t)}}$ that fits with the neutral expression T_n. For the query frame $Q_e^{(q)}$, we apply the inverse flow of $\boldsymbol{F}'_{Q_n\to Q_e^{(q)}}$ computed in Equation 4 on $\mathrm{d}\boldsymbol{F}_{Q_e^{(q)}}$, resulting in a warped velocity flow $\mathrm{d}\boldsymbol{F}'_{Q_e^{(q)}}$ that is also mapped to the neutral face T_n. Finally, the expression velocity difference between $Q_e^{(q)}$ and $T_e^{(t)}$ is computed as:

$$\begin{aligned} d_v(Q_e^{(q)}, T_e^{(t)}) = {} & \alpha_e \sum_{i\in\text{eye}} d_b(\mathrm{d}F'_{Q_e^{(q)},i}, \mathrm{d}F'_{T_e^{(t)},i}) \\ & + \alpha_m \sum_{i\in\text{mouth}} d_b(\mathrm{d}F'_{Q_e^{(q)},i}, \mathrm{d}F'_{T_e^{(t)},i}), \end{aligned} \tag{6}$$

where function $d_b(\cdot,\cdot)$ is defined in Equation 3. Combining Equation 4 and 6 together, the final expression distance metric for video is defined as:

$$\mathcal{D}(Q_e^{(q)}, T_e^{(t)}) = \gamma_e d_e(Q_e^{(q)}, T_e^{(t)}) + \gamma_v d_v(Q_e^{(q)}, T_e^{(t)}), \tag{7}$$

where $\gamma_{\{e,v\}} \in [0,1]$ are the weights for the expression distance and expression velocity distance, respectively, and subjected to $\gamma_e + \gamma_v = 1$.

Figure 4 shows an example which demonstrates that when the expression is subtle, incorporating the expression

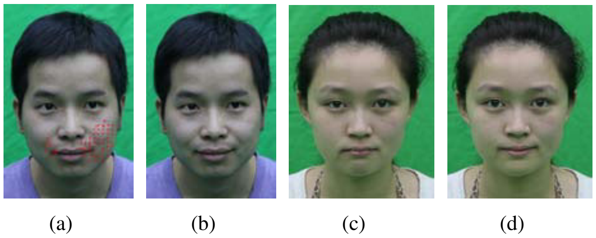


Figure 4. Illustration of the importance of using expression velocity. (a) Current query frame with expression velocity drawn in red. (b) The next query frame with a subtle smile. (c) The most similar expression chosen by the metric in Equation 4, which has a subtle sadness. (d) The most similar expression chosen by the metric in Equation 7, which has the right subtle smile.

velocity into the metric can help the system better capture the expression change.

4.2.2 Optimization-based Retrieval

The improved expression similarity metric alone cannot completely solve the temporal coherence problem. Our system thus employs an optimization-based retrieval approach to further improve the temporal coherence of the synthesized sequence.

For each query frame, we first extract not one, but k-nearest neighbors ($k = 20$ in our system) from the database, which we call the candidate frames, using the complete distance metric defined in Equation 7. By placing k candidate frames in a column at the timestamp of each frame, we build a directed acyclic graph as shown in Figure 5. Directed edges only connect candidates on adjacent frames. Let $V_i^{(q)}$ be the ith candidate at time q. We define the length (or cost) of the directed arc $r = (V_i^{(q)}, V_j^{(q+1)})$ as:

$$\begin{aligned}\mathcal{L}(r) = {} & \mathcal{D}(V_i^{(q)}, Q_e^{(q)}) + \mathcal{D}(V_j^{(q+1)}, Q_e^{(q+1)}) \\ & + \lambda \exp(-(\mathcal{T}(V_j^{(q+1)}) - \mathcal{T}(V_i^{(q)}) - \mu)^2/\sigma^2) \quad (8)\end{aligned}$$

where $\mathcal{T}(\cdot)$ indicates the timestamp of an input frame. By minimizing the timestamp difference of adjacent frames, the last term in Equation 8 encourages consecutive frames in the database to be chosen as the matched frames to maintain temporal coherence. The temporal scale variable μ is used to compensate for the difference in motion speed between the query and database sequences. μ is set to 1 if the query sequence has roughly the same motion speed as the database sequences, and is set to a larger number if the motion in the query sequence is faster than that of the database sequences, and vice versa. σ is the bandwidth and λ is the weight for the temporal term.

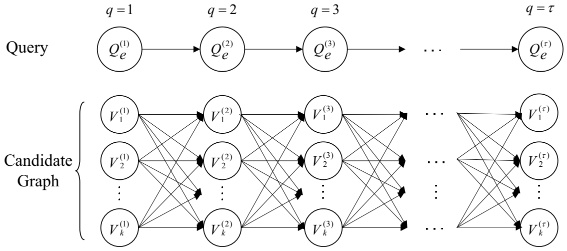


Figure 5. Directed graph for retrieval.

The temporal term in Equation 8 is a L2 norm which allows small timestamp shift, but penalizing heavily on large shift. Since small shifts are allowed, it allows certain amount of temporal scale changes as well. This is evidenced by our query sequence 1 shown in Figure 7, which contains both slow mouth-opening expression and fast pouting expression. Our system handles both cases well. In addition, μ can also be adjusted automatically at different time of the video according to speed of query motion.

Let $\mathcal{P}_{V_i^{(1)} \to V_j^{(\tau)}}$ be the path connecting the starting node $V_i^{(1)}$ and the ending node $V_j^{(\tau)}$, where $i, j \in \{1, 2, \dots, k\}$. Among all possible paths from the first to the last frame, we find a path with the shortest length. This optimization objective is formally formulated as:

$$\mathcal{P}_{opt} = \arg\min_{i,j} \arg\min_{\mathcal{P}_{V_i^{(1)} \to V_j^{(\tau)}}} \sum_{r \in \mathcal{P}_{V_i^{(1)} \to V_j^{(\tau)}}} \mathcal{L}(r). \quad (9)$$

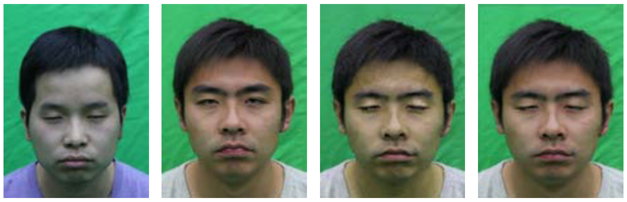
This problem can be effectively solved using Dijkstra's algorithm with Fibonacci heaps [6]. All frames connected by the optimal path $\mathcal{P}_{opt}$ form the retrieved sequence.

Our optimization-based formulation draws on prior work for creating temporal coherent animation [12, 13, 11]. Temporal coherence and semantic correspondence are both incorporated in our approach.

4.3. Expression Refinement

The previous retrieval result has two drawbacks. First, since the size of our database is limited, retrieved frames may not contain exactly the same expression as the input sequence. Second, database frames are not perfectly aligned, so the retrieved sequence contains some small amount of temporal jittering. To remove these artifacts, our system employs an additional expression refinement component.

The key idea of the expression refinement step is that given Q_n and Q_e, the neutral and expression frames of the puppeteer, and T_n, the neutral frame of the target person, we can directly extract the expression from the two source images and map it to T_n to synthesize a new face T_{Q_e}. This synthesized face, which we call the expression mapping image (EMI), has the desired expression, but may not have realistic texture, especially when the expression change between Q_n and Q_e is large. On the other hand, the retrieved frame has realistic appearance, but the expression does not



(a) (b) (c) (d)

Figure 6. Expression refinement. (a) Query frame. (b) Retrieved frame. (c) Expression mapping image. (d) Final result.

match with Q_e perfectly. Combining the EMI and the retrieved frame together, we can generate a final image that has both realistic appearance and accurate expression that matches well with the puppeteer.

Specifically, we first warp T_n by transferring the optical flow between Q_n and Q_e to the target frame. Given point $\vec{a} \in Q_n$, $\vec{a}' \in Q_e$, and $\vec{b} \in T_n$ ($\vec{b} = g(\vec{a})$ as in Equation 1), we compute the color of point $\vec{b}' \in T_{Q_e}$ as:

$$c_{\vec{b}'} = c_{\vec{b}} \frac{c_{\vec{a}'}}{c_{\vec{a}}}, \tag{10}$$

where $c_{\{\vec{a},\vec{a}',\vec{b},\vec{b}'\}}$ are the color values (YCrCb color space is used in our system) of the point $\vec{a}$, $\vec{a}'$, $\vec{b}$ and $\vec{b}'$, respectively. Here, we use the ratio $c_{\vec{a}'}/c_{\vec{a}}$ to modulate the color $c_{\vec{b}}$, as was done in the ERI approach [17]. In practical implementation, to avoid having a non-integer coordinate for $\vec{b}'$, we compute the expression mapping in the reversed direction. We start from an integer pixel $\vec{b}' \in T_{Q_e}$, and find its corresponding point $\vec{a}' \in Q_e$ as $\vec{a}' = g^{-1}(b')$. The flow $\Delta\vec{a}'$ obtained by calculating $\boldsymbol{F}_{Q_e \to Q_n}$ gives the position of point $\vec{a}$. Through the registration function, we obtain the position of point $\vec{b} = g(\vec{a}) \in Q_n$. Then the color of point $\vec{b}'$ can be calculated by Equation 10.

Finally, we compute the optical flow between the EMI and the retrieved result at each frame time, and warp the retrieved image towards the EMI result using the flow. As shown in Figure 6, the final synthesized frame has not only the realistic appearance inherited from the retrieved frame, but also the correct expression that matches well with the query frame, inherited from the EMI image.

5. Results and Discussion

5.1. Experiment Setup

We evaluate the system on three databases. Two of them were captured by ourselves, for which the two subjects, one male and one female, were asked to perform 6 basic expressions: anger, disgust, surprise, fear, happiness and sadness. Each database was captured at 25fps and consists of about 1500 frames. The third database is subject S130 from Extended Cohn-Kanade Dataset (CK+) [18]. 11 small sequences (220 frames in total) of subject S130 were pooled together to form the expression database. In all experiments the parameters in our algorithm were fixed as follows: $\alpha_e = \frac{0.6}{n_e}$, $\alpha_m = \frac{0.4}{n_m}$, $\beta_m = 0.9$, $\beta_o = 0.1$, $\gamma_e = 0.9$, $\gamma_v = 0.1$, $k = 20$, $\lambda = 0.1$, $\mu = 1$, $\sigma = 2$, where n_e and n_m are the numbers of pixels in the eye and mouth regions of the target neutral face, respectively.

5.2. Results and Evaluation

Figure 7 shows the synthesized results of the target subject $T1$ (male) and $T2$ (female), driven by an input sequence. Note that our system not only can synthesize realistic performance when the input expression is in the database, such as smile and surprise, but also can synthesize new expressions that are not covered in the database, such as a pouting mouth with closed eyes. Meanwhile, the final synthesized video is also temporally coherent. Figure 8 shows the synthesized result of subject S130 from CK+ database, driven by another sequence. Results show that our system works well even with a small database. The complete set of video, including an additional fast talking retargeting result, can be found in the supplementary material.

To evaluate the quality of our synthesized results, we performed a user study with 34 participants. Each participant was shown four videos obtained by frame-by-frame query approach using LBP features in [10], EMI introduced in Section 4.3, our retrieval strategy, and our entire algorithm. Each video presents the query and a result side by side. In the experiment, participants were asked to rate how good the facial expression in each result looks based on realism and consistency with the driving performance, on a scale from 5 (very good) to 0 (not good at all). Table 1 states the average scores for 3 target subjects. Participants found that our final result is the best one and our retrieval strategy outperforms the method proposed in [10].

	$T1$	$T2$	S130
LBP-based retrieval [10]	1.20	1.50	1.38
Our retrieval	2.49	3.00	2.56
EMI	2.89	1.91	3.35
Our entire system	**4.02**	**4.56**	**4.08**
p-value	0.002	0.005	0.0001

Table 1. User study results. The results are all statistically significant with the one-way ANOVA p-value < 0.01.

5.3. Limitations

Our current system is designed only for frontal facial expression synthesis. Extending the system to work under large rotation angles is possible, by capturing the database of the target person using a sparse camera array. In this case we need to estimate both the expression and the 3D pose of the face from the input frames. Furthermore, view morphing technique [24] is needed to interpolate faces be-

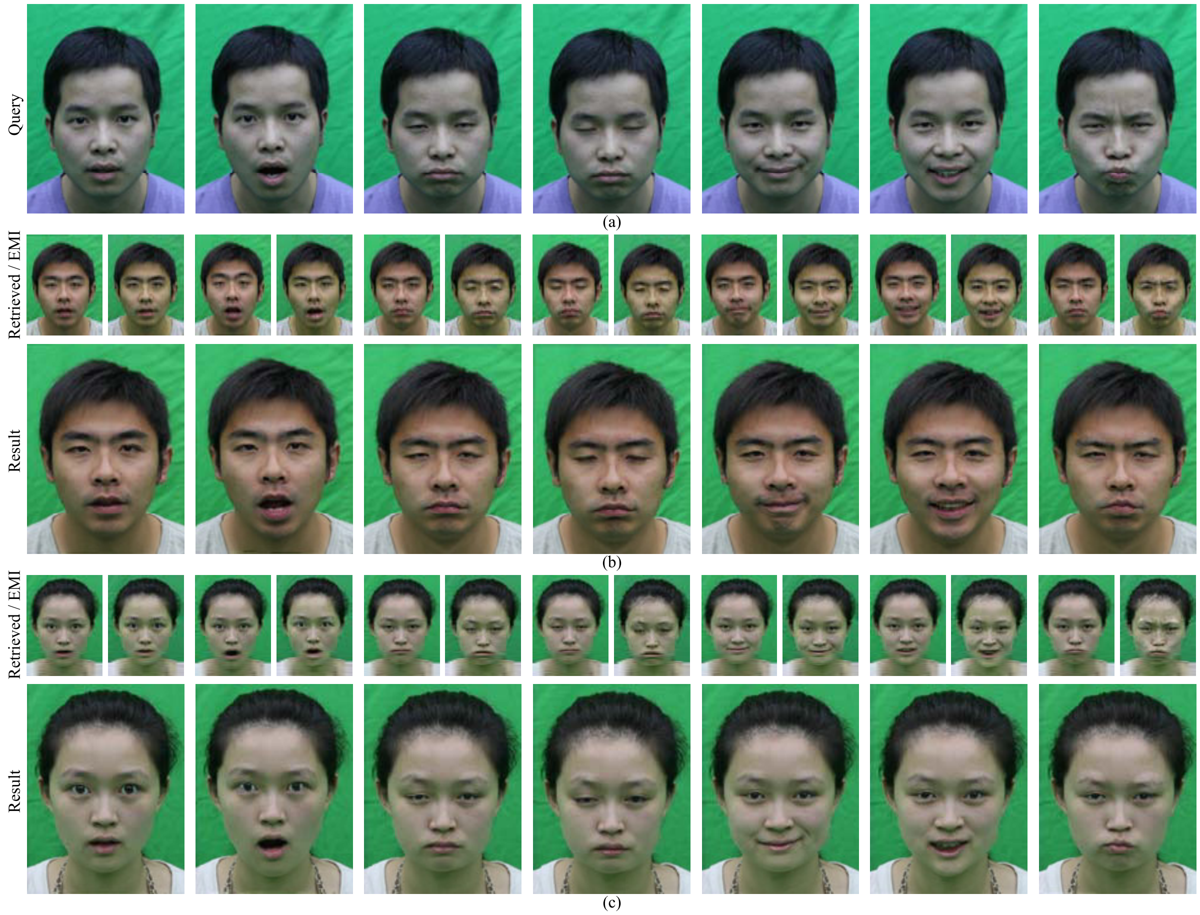


Figure 7. Result of query sequence 1 performed on target $T1$ and $T2$. (a) Common query frames. (b) (first row) Retrieved and EMI frames (left and right, respectively); (second row) Final synthesized frames of target $T1$. (c) Retrieved, EMI, and final frames of target $T2$.

tween different viewpoints to generate facial expressions at desired poses.

Another limitation is that when the expression is extreme, traditional optical flow methods cannot accurately capture the expression difference. Besides investigating better facial flow techniques, another solution is to use multiple pre-aligned anchor facial images for each character instead of using one neutral face in our current system.

6. Conclusion

We proposed a data-driven approach for synthesizing a facial animation of a target person driven by a facial performance video of another person. Our system employs a novel spatial-temporal expression distance metric that can accurately measure expression similarity of different people in video. We also propose a shortest path optimization-based retrieval strategy to balance between expression similarity and temporal smoothness in the final video. The expression similarity is further improved by warping the retrieved video frames towards those created by direct expression mapping. User study results have shown that our system can generate high fidelity and temporally-coherent facial animation.

Acknowledgments

The authors would like to thank Ruiping Wang, Yue Deng, and Jinli Suo for helpful discussions, and the reviewers and area chairs for constructive comments. This research is supported by the Project of NSFC (No. 61035002, No. 61073072, and No. 60933006).

References

[1] T. Beier and S. Neely. Feature-based image metamorphosis. In *SIGGRAPH*, pages 35–42, 1992.

[2] V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating faces in images and video. *Computer Graphics Forum*, 22(3):641–650, 2003.

[3] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, pages 187–194, 1999.

[4] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE TPAMI*, 23(6):681–685, 2001.

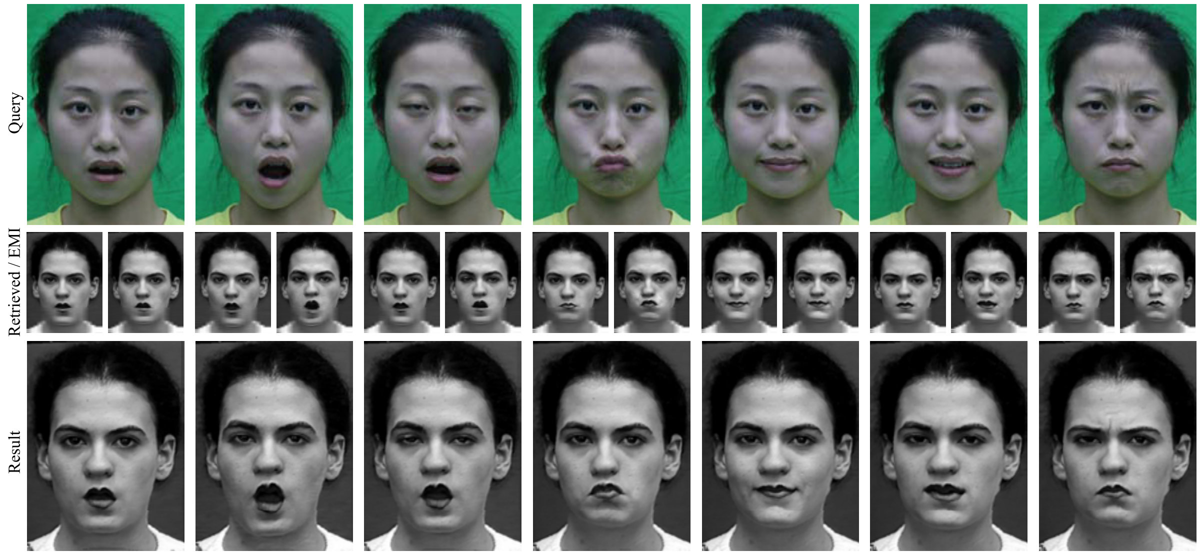


Figure 8. Result of query sequence 2 performed on subject S130 from CK+. (top) Query frames. (middle) Retrieved and EMI frames (left and right, respectively). (bottom) Final synthesized frames.

[5] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[6] T. H. Cormen, C. E. Leiserson, R. L. Riverst, and C. Stein. *Introduction to algorithms*. MIT press, Cambridge, 2009.

[7] K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister. Video face replacement. *ACM Trans. Graph.*, pages 130:1–130:10, 2011.

[8] Y. Deng, Q. Dai, and Z. Zhang. Graph laplace for occluded face completion and recognition. *IEEE Trans. Image Process.*, 20(8):2329 –2338, 2011.

[9] P. Ekman and W. V. Friesen. *Facial action coding system: a technique for the measurement of facial movement*. Consulting Psychologists Press, Palo Alto, 1978.

[10] I. Kemelmacher-Shlizerman, A. Sankar, E. Shechtman, and S. M. Seitz. Being john malkovich. In *ECCV*, pages 341–353, 2010.

[11] I. Kemelmacher-Shlizerman, E. Shechtman, R. Garg, and S. M. Seitz. Exploring photobios. *ACM Trans. Graphics*, pages 61:1–61:10, 2011.

[12] L. Kovar and M. Gleicher. Flexible automatic motion blending with registration curves. In *SCA*, pages 214–224, 2003.

[13] Z. Li, N. Snavely, B. Curless, and S. M. Seitz. Spacetime faces: high resolution capture for modeling and animation. *ACM Trans. Graph.*, 23(3):548–558, 2004.

[14] G. Littlewort, J. Whitehill, T. Wu, I. Fasel, M. Frank, J. Movellan, and M. Bartlett. The computer expression recognition toolbox (cert). In *FG*, pages 298–305, 2011.

[15] P. Litwinowicz and L. Williams. Animating images with drawings. In *SIGGRAPH*, pages 409–412, 1994.

[16] C. Liu. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, MIT, 2009.

[17] Z. Liu, Y. Shan, and Z. Zhang. Expressive expression mapping with ratio images. In *SIGGRAPH*, pages 271–276, 2001.

[18] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *CVPR Workshops*, pages 94–101, 2010.

[19] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba. Coding facial expressions with gabor wavelets. In *FG*, 1998.

[20] J.-y. Noh and U. Neumann. Expression cloning. In *SIGGRAPH*, pages 277–288, 2001.

[21] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE TPAMI*, 24(7):971–987, 2002.

[22] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin. Synthesizing realistic facial expressions from photographs. In *SIGGRAPH*, pages 75–84, 1998.

[23] J. M. Saragih, S. Lucey, and J. F. Cohn. Real-time avatar animation from a single image. In *FG*, pages 117–124, 2011.

[24] S. M. Seitz and C. R. Dyer. View morphing. In *SIGGRAPH*, pages 21–30, 1996.

[25] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. *ACM Trans. Graph.*, 24(3):426–433, 2005.

[26] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. *ACM Trans. Graph.*, pages 77:1–77:10, 2011.

[27] T. Weise, H. Li, L. Van Gool, and M. Pauly. Face/off: live facial puppetry. In *SCA*, pages 7–16, 2009.

[28] L. Williams. Performance-driven facial animation. In *SIGGRAPH*, pages 235–242, 1990.

[29] F. Xu, Y. Liu, C. Stoll, J. Tompkin, G. Bharaj, Q. Dai, H.-P. Seidel, J. Kautz, and C. Theobalt. Video-based characters: creating new human performances from a multi-view video database. *ACM Trans. Graph.*, pages 32:1–32:10, 2011.

[30] Q. Zhang, Z. Liu, B. Quo, D. Terzopoulos, and H.-Y. Shum. Geometry-driven photorealistic facial expression synthesis. *IEEE TVCG*, 12(1):48–60, 2006.