FACIAL EXPRESSION TRANSFER BY USING 3D-AWARE EXPRESSION FLOW

BY FEI YANG

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Professor Dimitris N. Metaxas

and approved by

____

____

____

____

____

New Brunswick, New Jersey

OCTOBER, 2013

ABSTRACT OF THE DISSERTATION

Facial Expression Transfer by Using 3D-Aware Expression Flow

by Fei Yang

Dissertation Director: Professor Dimitris N. Metaxas

We study the problem of transferring facial expressions from one face to another. Direct copying and blending face components using existing methods results in semantically unnatural composites, since expression is a global effect. A local face component in one expression is often incompatible with the shape and other components in another expression. To solve this problem we present the expression flow method, which is a 2D flow field that can warp the target face globally. We develop a shape fitting algorithm, which jointly constructs 3D face shapes to the input images with the same identity but different expressions. The expression flow is computed by projecting the difference between the two 3D shapes to 2D image plane. We apply our algorithms in several applications including face compositing, face morphing, video stitching, and facial expression exaggeration. Our system is able to generate faces with much higher fidelity than existing methods.

Acknowledgements

I am grateful to Professor Dimitris Metaxas for his advice and support during my PhD study. He has been motivating and encouraging me to solve fundamental problems in Computer Vision, Computer Graphics and Machine Learning. None of the work in this dissertation would have happened without him.

I would like to thank other members of my doctoral committee: Prof. Tina Eliassi-Rad, Prof. Kostas Bekris, and Prof. Xiaolei Huang for their advices, help and valuable suggestions regarding this dissertation. It is a honor for me to have each of them serves in my committee.

I also thank Dr. Jue Wang, Dr. Eli Shethman and Dr. Lubomir Bourdev from Adobe Creative Labs, Dr. Hong Jiang from Bell Labs, and Prof. Junzhou Huang from University of Texas at Arlington. They have been my mentors, and gave me a lot of help in my research.

Special thanks to all my colleagues and fellow students at the Center for Computational Biomedicine Imaging and Modeling (CBIM) and the Computer Science Department at Rutgers University. I benefited a lot from working with and learning from these smart people.

Dedication

To my wife Zao Zhang, and my parents Jichen Yang and Li Guo.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Everyone who has the experience of taking photographs of family members and friends knows how hard it is to capture the perfect moment. For one, the camera may not be at the right setting at the time. Furthermore, there is always a delay between the time one sees a perfect smile in the viewfinder and the time that the image is actually captured, especially for low-end cell phone cameras which have slow response. For these reasons, face images captured by amateur photographers often contain various imperfections. Generally speaking, there are two types of imperfections. The first type is photometric flaws due to improper camera setting. The face may appear to be too dark, grainy, or blurry. The second type, which is often more noticeable and severe, is the bad expression of the subject, such as closed eyes, half-open mouth, etc.

With recent advances in image editing, photometric imperfections can be largely improved by using modern photo editing software. For instance, the personal photo enhancement system [54] provides a set of adjustment tools to correct global attributes of the face such as color, exposure, and sharpness. Compared with photometric imperfections, expression artifacts are much harder to correct. Given a non-smiling face photo, one could simply find a smiling photo of the same person from his/her personal album, and use it to replace the whole face via existing methods [12]. Unfortunately, this global swap also replaces other parts of the face which the user may want to keep. Local component transfer among face images is thus sometimes more preferable.

In addition to exploding amount of photos, videos are also captured at astonishing rates. In 2011, on YouTube alone, every minute people upload 8 years of video content. With respect to video capturing, bandwidth and storage have become easier over time, semantic editing of video content remains a very challenging problem. Although a

Photoshop expert with sufficient amount of time could change one's expression in an image, doing so in video is prohibitively expensive. This is because the time dimension adds new sets of constraints, such as temporal coherence, and the temporal "signature" of an expression.

In this dissertation, we aim to develop a framework that could automatically edit facial expression in photos and videos in a semantic level. The proposed method is non-intrusive, computationally efficient, and can be used with existing photos or videos. Our system consists of four major components: facial feature localization, face model fitting, expression flow computation, and image compositing. We will use one chapter to address facial feature localization, and then introduce three different applications: facial component transfer, expression editing, and face morphing. For each application, we will introduce its own model fitting, image warping and composition algorithms.

In Chapter 2, we first review existing facial feature localization algorithms. We propose a new shape registration method, which explicitly models the registration error of occluded landmarks. We extend the linear subspace shape model by introducing a misplacement term of the occluded landmarks. We assume that the landmark misplacement term is a sparse vector, as long as the occlusion takes a small part on the face. With the sparse misplacement term involved in shape searching procedure, our method is more robust under partial occlusion.

We also introduce a new eye localization method via Multiscale Sparse Dictionaries. We build a pyramid of dictionaries that models context information at multiple scales. Eye locations are estimated at each scale by fitting the image through sparse coefficients of the dictionary. By using context information, our method is robust to various eye appearances.

In Chapter 3, we address the problem of correcting an undesirable expression on a face photo by transferring local facial components from another face photo of the same person which has the desired expression. To make the target face compatible with the new facial component, we present Expression Flow, a 2D flow field which can warp the target face globally in a natural way. Starting with the two input face photos, we jointly construct a pair of 3D face shapes with the same identity but different expressions. The expression flow is computed by projecting the difference between the two 3D shapes back to 2D. It describes how to warp the target face photo to match the expression of the reference photo. User studies suggest that our system is able to generate face composites with much higher fidelity than existing methods.

In Chapter 4, we address the problem of editing facial expression in video, such as exaggerating, attenuating or replacing the expression with a different one. 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. 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.

In Chapter 5, we propose a new face morphing approach that deals explicitly with large pose and expression variations. We recover the 3D face geometry of the input images using a projection on a prelearned 3D face subspace. The geometry is interpolated by factoring the expression and pose and varying them smoothly across the sequence. Finally we pose the morphing problem as an iterative optimization with an objective that combines similarity of each frame to the geometry-induced warped sources, with a similarity between neighboring frames for temporal coherence. Experimental results show that our method can generate higher quality face morphing results for more extreme pose, expression and appearance changes than previous methods.

In Chapter 6, we conclude our work.

Chapter 2

Facial Feature Localization

Accurately detecting and tracking facial features plays an important role in many applications, such as face recognition and human behavior analysis. It is a highly challenging task. Different view angles, expressions and lighting conditions greatly alter the face appearances and increase the complexity of the problem. Facial landmarks are generally defined at mouth corners, eye corners, or placed evenly at the boundaries of cheeks and facial features (see Fig. 2.1). In this chapter, we first review existing methods for facial landmark localization and tracking. Then we propose a new approach to localize facial landmarks for partially occluded faces. We also propose a method to localize eye centers based on sparse dictionary reconstruction.

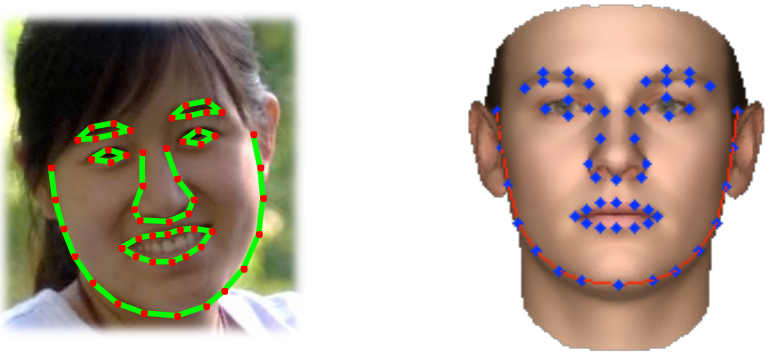

Figure 2.1: Facial landmarks and their corresponding 3D locations.

2.1 Related work

Face models and facial landmark localization have been studied extensively in both computer vision and computer graphics communities. The existing approaches can be categorized into parametric models, active shape models, constrained local models, and active appearance models. Based on recent advances of consumer depth cameras, depth data has also been utilized in model fitting and landmark localization.

Parametric models

The early work of facial feature localization is limited to either rigid head motion with static expressions [4, 5], or varying expressions of a roughly stationary head [113, 93]. Parametric face models were explored in 1990's to detect and track facial features. These models are carefully designed with a set of parameters controlling the deformations driven by elastic forces or image motion. Black et al. [13, 14] explored local parameterized models and used image motion for recovering and recognizing non-rigid motion of human faces. De Carlo et al. [35, 36] modeled 3D face with a polygon mesh, and applied optical flow as a non-holonomic constraint. Pighin et al. [77] proposed to model 3D face as a linear combination of several texture-mapped models, each corresponding to a particular basic facial expression. A scattered data interpolation approach was proposed to deform the model to fit faces from photographs.

In the above-mentioned approaches, the models and parameters are carefully defined by experts and animators based on their experiences. In contrast, statistical approaches learn face models from a set of training faces of various identities and expressions [17, 15]. The training data are usually captured as high accuracy 3D scans, which represent a wide variety of faces and facial motions. The Active Shape models, Constrained Local Models and Active Appearance models introduced in the following sections are all statistical based models.

Active shape models

The Active Shape Models (ASM) [27] capture the statistical distributions of 2D feature points, thus allowing shapes to vary only in the ways seen in a training set. ASM models have been successfully used in many shape fitting problems. Kanaujia and Metaxas [55] built a real-time face tracking system based on ASM. They trained a mixture of ASMs, each corresponding to a different pose. The target shape is fitted by first searching the local features along the normal direction, followed by constraining the global shape using the most probable cluster.

2D ASM based methods are also combined with 3D face models, which govern the overall shape, orientation and location. Vogler et al. [100] developed an framework to integrate both 2D ASM and 3D deformable models, which allows robust tracking of faces and estimating both rigid and non-rigid motions. The displacements between the actual projected model points and the identified correspondences are defined as image forces to update the deformation parameters. Yang et al. [112] proposed a face fitting method by combining statistical models of both 2D and 3D faces. Shape fitting was performed by minimizing both feature displacement errors and subspace energy terms with temporal smoothness constraints.

Given limited number of training samples, traditional statistical shape models may over fit and generalize poorly for new samples. Instead of building models on the entire face, Huang et al. [52] built separate ASM models for face components to preserve local shape deformations. Markov Network is used to set global geometry constraints. Some recent research works enhanced the ASM fitting by using sparse displacement errors [110, 117]. These models are more robust to outliers and partial occlusions.

Constrained local models

The Constrained Local Models (CLM) are extension of Active Shape Models. They use an independent set of local detectors for landmark detection [33]. CLM fitting is generally posed as the search for the point distribution parameters that jointly minimizes the misalignment error over all landmarks. The misalignment error of each landmark measures the distance of the current shape from the shape distribution, which is often modeled as Gaussian [9] or mixture of Gaussian (GMM) [47]. Examples of the misalignment error functions include the Mahalanobis distance for local patch appearances [27], or the output of feature detectors [33].

As local landmark detectors are learned from small image regions with limited structures, the maximum responses may not coincide with the correct landmark locations. Some recently proposed methods try to alleviate this problem. Wang et al.[102] proposed a convex quadratic function to fit to the negative log of the response map, from which the mean and covariance of the approximating density can be inferred. Zhou et al. [120] used the summed-squared-difference as a measure of landmark fit, and applied Laplace's approximation to find the covariance estimate. Saragih et al. [82, 83] proposed an optimization strategy where a nonparametric representation of the landmark distributions is maximized within a hierarchy of smoothed estimates. The resulting update equations are reminiscent of mean-shift but with a subspace constraint placed on the shape's variability.

Active appearance models

The ASM and CLM introduced above only model statistical distributions of the shapes. In contrast, Active appearance models (AAM) decouple the shape and texture of the deformable object, and are able to generate a variety of photo-realistic instances [30]. Fitting an AAM to an image consists of minimizing the error between the input image and the closest model instance, i.e. solving a nonlinear optimization problem. Matthews et al. [69] suggested to reformulate AAM fitting as an image alignment problem [6], which can be efficiently solved by LucasKanade inverse compositional algorithm [7]. The proposed method avoids updating texture parameters and turns out to be the fastest fitting algorithm for AAM.

AAM has been successfully used for real-time face tracking. To deal with pose variations, Cootes et al. [31] proposed view-based AAM models, which is a combination of a few 2D models. Sung et al. [89] combined AAM with a cylinder head model, where the global head motion parameters obtained from the cylinder model are used as the cues of the AAM parameters for a good fitting or re-initialization. Xiao et al. [109, 70] proposed a real time face tracking algorithm by combining 2D+3D AAM models. Zhou et al. [119] introduced temporal matching constraints to enforce inter-frame coherence in AAM fitting. A comprehensive review of AAM models is provided by Gao et al.[45].

Face alignment based on range data

For optical cameras, severe lighting conditions may significantly alter the appearance of features and cast shadows on faces. In contrast, range data is insensitive to environment lighting. Structured light based methods were first explored for capturing depth maps of moving faces [78, 51]. Zhang et al. [116] developed a 3D face tracking system by employing synchronized video cameras and structured light projectors to capture streams of images from multiple viewpoints. The 3D shapes were matched to a template by using both depth error and shape regularization.

Based on recent development of consumer depth cameras, using range data in face tracking has received much attention. Microsoft released a consumer depth camera - Kinect in November 2010. Fanelli et al. [41] developed an algorithm which uses random forests to estimate head orientations from the range data. Cai et al. [20] developed a maximum likelihood solution to track face shapes from the noisy depth images. Weise et al. [103] developed a realtime system to track face geometry from a depth camera, and created face animations by applying the tracked 3D motion to cartoon characters. Baltrusaitis et al. [8] extended the Constrained Local Models to use both depth and intensity images for face tracking. Microsoft also released the official Kinect SDK, which contains functions to track facial landmarks and head motions in real time by using a Kinect camera [118].

2.2 Facial feature localization with occlusion

Previous landmark localization methods (e.g. ASM) perform poorly when the face is partially occluded. This is because the occluded landmarks cannot be matched to the correct positions. While the incorrect matches are projected into the shape space with the correct ones, which leads to a distorted shape. Some previous methods are claimed to be more robust in such cases. For example, Bayesian inference based on tangent shape approximation[121], generative model with EM-based algorithm to compute the maximum a posterior[47], and pictorial structures which model the spacial relationship between parts of objects[44, 91]. However, facial feature localization under occlusion is far from being solved.

In this section, we propose a new shape registration method, which explicitly models the registration error of occluded landmarks. We extend the linear subspace shape

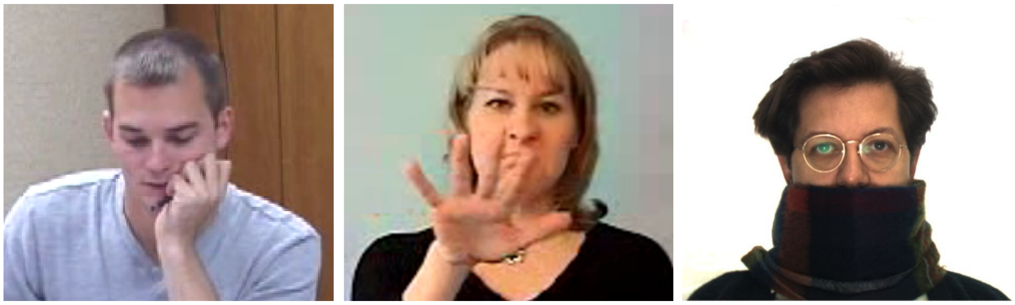

Figure 2.2: Facial feature localization is very challenging when faces are partially occluded. A face could be occluded by hand touching (left), hand gesture making (middle), or facial accessories (right).

model by introducing an misplacement term of the occluded landmarks. We assume that occlusion takes a small part of the face, so the landmark misplacement term is a sparse vector. The proposed method iteratively approximates the optimal shape. To quantitatively evaluate the proposed method, we built three face datasets with synthesized occlusions. Our experimental results prove the advantage of our method.

2.2.1 Sparse shape registration

Given a shape containing N landmarks, the shape vector S is defined by concatenating x and y coordinates of all the landmarks.

$$S = [x_1, y_1, x_2, y_2, ..., x_N, y_N]^T \tag{2.1}$$

We assume the shape is a linear combination of m shape basis

$$\begin{aligned} S &= \bar{S} + b_1 u_1 + b_1 u_1 + \cdots + b_m u_m && (2.2) \\ &= \bar{S} + Ub && (2.3) \end{aligned}$$

where U is a n by m matrix which contains m shape basis. b is a m by 1 vector containing shape coefficients. If a facial landmark is occluded, it may not get a high response at its correct location. And the position with highest response in the search window may not be the real position at all. Therefore, such incorrect position should not be used for global shape matching.

We define an error term S_e to directly model the occluded landmark positions. The hidden shape vector S is the sum of the shape estimate $\hat{S}$ and shape error S_e.

$$S = \hat{S} + S_e \tag{2.4}$$

The shape transformation parameters (scaling, rotation and translation) are denoted by θ. The posterior likelihood of θ, shape parameter b, hidden shape vector S, error S_e given image I is:

$$p(\theta, b, S, S_e | I) \propto p(\theta)p(b)p(S|b)p(S_e)p(I|\theta, S, Se) \tag{2.5}$$

The prior $p(\theta)$ can be considered as a constant, since there is no preference for shape scale, orientation and location. We take the negative logarithm of Equation (2.5). Now we aim to minimize the following energy function:

$$\begin{aligned} E &= -\log p(b) - \log p(S|b) - \log p(S_e) - \log p(I|\theta, S, Se) && (2.6) \\ &= E_b + E_S + E_{S_e} + E_I && (2.7) \end{aligned}$$

The subspace energy term E_b is defined as:

$$E_b = \frac{1}{2} b^T \Lambda^{-1} b \tag{2.8}$$

where Λ is the m by m diagonal matrix containing the largest m eigenvalues of Σ. For simplicity, we consider the shape model to be a single Gaussian distribution with mean $\bar{S}$ and covariance Σ^{-1}. The shape basis U and Λ are computed from SVD decomposition of covariance matrix $\Sigma = U\Lambda U^T$. The shape basis is defined by the m eigenvectors corresponding to the m largest eigenvalues of Σ. The single Gaussian model can also be extended to a mixture of Gaussians[47].

The shape energy E_S can be written as

$$\begin{aligned} E_S &= \frac{1}{2}||S - Ub - \bar{S}||^2 && (2.9) \\ &= \frac{1}{2}||\hat{S} + S_e - Ub - \bar{S}||^2 && (2.10) \end{aligned}$$

We assume that the occlusion only takes a small part on the face, which means S_e is sparse. We define the energy term E_{S_e} as the L_1 norm of S_e, with a diagonal weighting matrix W.

$$E_{S_e} = \lambda \cdot ||WS_e||_1 \tag{2.11}$$

The image likelihood at each landmark position is assumed to be independent to each other. So that

$$p(I|\theta, S, Se) = \prod_{i=1}^{N} p(I_i|\theta, S, S_e) \tag{2.12}$$

We also use a single Gaussian model for the appearance at each landmark position. Thus the energy term E_I can be written as

$$\begin{aligned} E_I &= \frac{1}{2}\sum_{i=1}^{N}(F(\mathbf{x}_i) - u_i)^T \Sigma_i^{-1}(F(\mathbf{x}_i) - u_i) && (2.13) \\ &= \frac{1}{2}\sum_{i=1}^{N} d(\mathbf{x}_i)^2 && (2.14) \end{aligned}$$

where $F(\mathbf{x}_i)$ is the feature extracted at landmark position $\mathbf{x}_i$ from shape $\hat{S}$; u_i and Σ_i are the mean and covariance of the Gaussian appearance model for landmark i. The energy term can be simply written as a sum of Mahalanobis distances $d(\mathbf{x}_i)^2$.

2.2.2 Iterative optimization

Now we aim to minimize the energy function E:

$$E = E_b + E_S + E_{S_e} + E_I \tag{2.15}$$

Algorithm 1 *Minimize* $Ep = E_b + E_S + E_{S_e}$

```
$b^0 = U^T(\hat{S} - \bar{S})$, $S_e^0 = 0$
for $k = 0 : k_{max}$ do
    Compute $L$ to be the largest eigenvalue of $\frac{\partial^2 E_p}{\partial b^2}$.
    $b^{k+1} = b^k - \frac{1}{L} \cdot \frac{\partial E_p}{\partial b}$
    $S_e^{k+\frac{1}{2}} = S_e^k - \frac{\partial E_p}{\partial S_e}$
    $S_e^{k+1} = max(|S_e^{k+\frac{1}{2}}| - \lambda, 0) \cdot sign(S_e^{k+\frac{1}{2}})$
end for
```

Firstly, we define E_p as the sum of E_b, E_S and E_{S_e}

$$E_p(b, S_e) = \frac{1}{2} b^T \Lambda^{-1} b + \frac{1}{2} ||\hat{S} + S_e - Ub - \bar{S}||^2 + \lambda \cdot ||W S_e||_1 \tag{2.16}$$

E_p is a convex function, which can be minimized by gradient descent method. The first and second order partial derivatives of E_p to b and S_e are:

$$\frac{\partial(E_b + E_S)}{\partial b} = \Lambda^{-1} b - U^T(\hat{S} + S_e - Ub - \bar{S}) \tag{2.17}$$

$$\frac{\partial(E_b + E_S)}{\partial S_e} = \hat{S} + S_e - Ub - \bar{S} \tag{2.18}$$

$$\frac{\partial^2 E_p}{\partial b^2} = (\Lambda^{-1} + I) \tag{2.19}$$

$$\frac{\partial^2 E_p}{\partial S_e^2} = I \tag{2.20}$$

The algorithm to minimize E_p is shown in Algorithm 1.

Secondly, we try to minimize E_I. Notice that E_I is a discontinuous function. Traditional active shape model based algorithms measure the image likelihood around the the landmarks, and move the landmark to the new position which has maximum response. For real world images, this method is sensitive to noises. Instead of using the single maximum response point, we use the kernel density estimation and mean shift method to find the position best matching the landmark.

Image gradient features are extracted at a set of n points $\{\mathbf{x}_{i,j}\}_{j=1\ldots n}$ around a landmark at point $\mathbf{x}_i$. we define $f(\mathbf{x}_{i,j})$ as the square of Mahalanobis distance at point

Algorithm 2 *Minimize E_I*

```
for $i = 1 : N$ do
    for $k = 0 : k_{max}$ do
        Compute $\hat{\nabla} f_{h,K}(\mathbf{x}_i^k)$ using equation (2.23)
        $\mathbf{x}_i^{k+1} = \mathbf{x}_i^k - \hat{\nabla} f_{h,K}(\mathbf{x}_i^k)$
    end for
end for
```

$\mathbf{x}_{i,j}$.

$$f(\mathbf{x}_{i,j}) = d(\mathbf{x}_{i,j})^2 \tag{2.21}$$

The kernel density estimation computed in the point $\mathbf{x}$, with kernel K and bankwidth h, is given by

$$\hat{f}_{h,K}(\mathbf{x}) = \frac{1}{C} \sum_{j=1}^{n} f(\mathbf{x}_{i,j}) \cdot K(\frac{\mathbf{x} - \mathbf{x}_{i,j}}{h}) \tag{2.22}$$

Let G be profile of kernel K. When K is the normal kernel, its profile G has the same expression. As shown in [25], the gradient estimate at point $\mathbf{x}$ is proportional to the density estimate in $\mathbf{x}$ computed with kernel G and the mean shift vector computer with kernel G.

$$\hat{\nabla} f_{h,K}(\mathbf{x}) = C \cdot \hat{f_{h,G}}(\mathbf{x}) \cdot m_{h,G}(\mathbf{x}) \tag{2.23}$$

The mean shift vector $m_{h,G}(\mathbf{x})$ is defined as

$$m_{h,G}(\mathbf{x}) = \frac{\sum_{j=1}^{n} \mathbf{x}_{i,j} \cdot f(\mathbf{x}_{i,j}) \cdot G\left(||\frac{\mathbf{x}-\mathbf{x}_{i,j}}{h}||^2\right)}{\sum_{j=1}^{n} f(\mathbf{x}_{i,j}) \cdot G\left(||\frac{\mathbf{x}-\mathbf{x}_{i,j}}{h}||^2\right)} - \mathbf{x} \tag{2.24}$$

The local minimum of E_I can be acquired using gradient descent. We take steps proportional to the negative of the gradient, as shown in Algorithm 1.

To minimize E, we alternately run Algorithm 1 and Algorithm 2. Our algorithm is shown in Algorithm 3.

Algorithm 3 *Sparse Shape Optimization*

```
Compute $\theta$ using detection result
Initial status $b_0 = 0$, $S_e = 0$, $S = \bar{S}$, $\hat{S} = \bar{S}$, $\hat{S}' = M_\theta(\hat{S})$
repeat
    Run Algorithm 2 to optimize $\hat{S}'$
    Compute transformation parameter $\theta$ matching $\hat{S}'$ to $\bar{S}$
    $\hat{S} = M_\theta^{-1}(\hat{S}')$
    Run Algorithm 1 to optimize $b$ and $S_e$
    $\hat{S}' = M_\theta(\bar{S} + Ub)$
until $\hat{S}'$ converges.
```

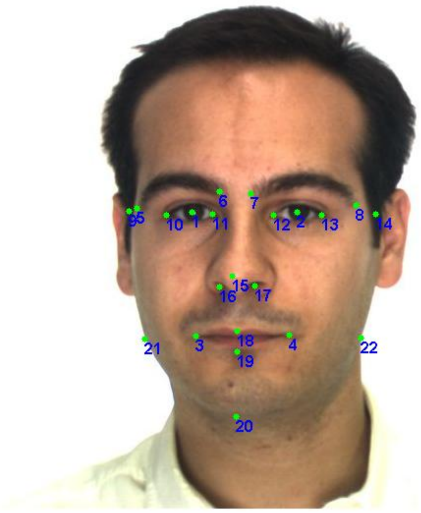

Figure 2.3: An example face image in AR database with 22 annotated landmarks.

2.2.3 Experiment

To evaluate our algorithm, we create a synthesized face occlusion database using face images from AR [68] database. The AR database contains frontal face images of 126 subjects. Each subject has 26 images with different expressions, occlusions and lighting conditions. We select 509 face images from section 1,2,3,5 and use the 22 landmark positions provided by T.F.Cootes [28] as the ground truth. The landmark positions are shown in Figure 2.3.

The occlusion masks are designed to simulate the occlusions most frequently seen

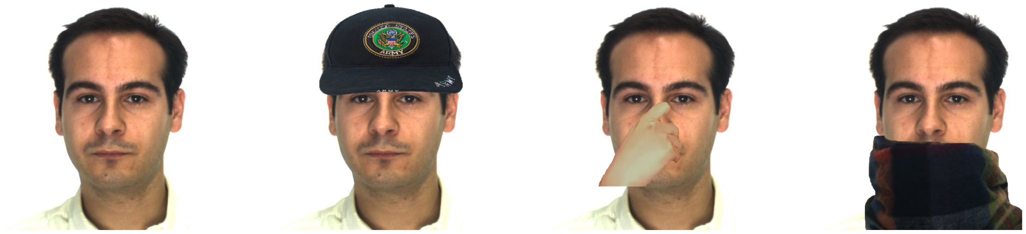

Figure 2.4: Faces with artificial occlusion

in real world. As shown in Figure 2.4. We design three types of masks. A hat is placed above the eyes and occludes all eye brow regions. A hand is placed on mouth and also occludes nose tip. And a scarf is placed to occlude the mouth and chin. These masks are carefully placed at the same position on all the faces. In this way, we still have ground truth positions of all occluded landmarks, which allows us to perform quantitative evaluations.

The shape registration result for one testing image is shown in Figure 2.5. The ground truth positions are marked using red stars. The result of ASM is shown in blue lines, and the result of our method is shown in green lines. On the right side is the sparse shape error recovered during one iteration. The non-zero coefficients on the left side corresponds to the landmarks at contour of the face. And the non-zero coefficients on the right side corresponds to the landmarks in mouth region. In this figure, we use a linear shape model containing 66 landmarks, and the shape error term S_e is of dimension 132.

In order to quantitatively evaluate the localization precision, we apply the normalized error metric similar to Jesorsky et al. [53]. The normalized error for each point is defined as the Euclidean distance from the ground truth, normalized by the distance between eye centers. Therefore, this metric is invariant to scale of faces.

We compare our algorithm with the Milborrow's extended Active Shape Model [71], which achieved better performance than traditional ASM methods. The results are shown in Figure 2.6. For the hat occlusion dataset, our method performs significantly better for landmarks 6,7,8,9, which are the four landmarks at the ends of eye brows. For the hand occlusion dataset, our method has much better accuracy for landmarks

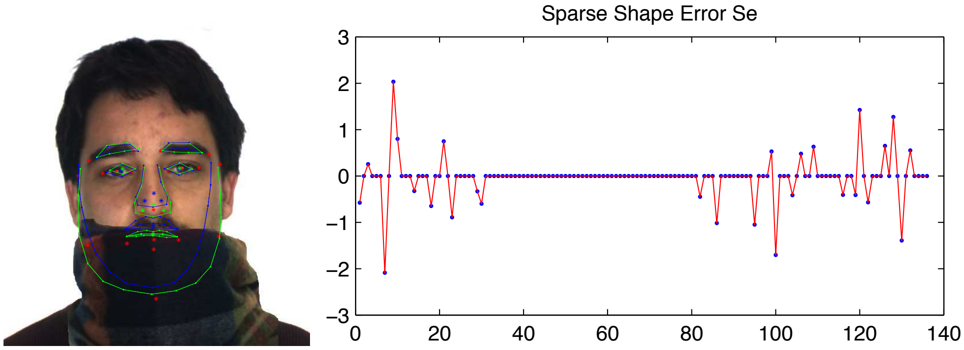

Figure 2.5: **Left:** The shape registration result of the proposed method (green) compared with EASM (blue). The ground truth positions are shown in red dots. **Right:** The sparse shape error term. The non-zeros values correspond to occluded landmarks.

3,4,18,19 which are occluded landmarks on mouth, and landmarks 15, 16, 17 which are occlude landmarks on nose. On the scarf occlusion dataset, our method gets much better accuracy for landmarks 3,4,18,19 which are occluded landmarks on mouth, and landmarks 20,21,22 which are occluded landmarks on chins.

In all the three datasets, we decrease the normalized error of the occluded landmarks to level of 0.2, which is close to the error level of the non-occluded landmarks. The speed of our method is about 20 percent slower than Milborrow's extended ASM, because of extra cost to compute gradients. We set the maximum number of iterations k_{max} to be 2 in Algorithm 1 and 2. Our experiments with larger k_{max} do not have significant better accuracy. We show more localization results in Figure 2.7. The ground truth positions are marked as red stars. The ASM results are shown in blue lines and our results are shown in green lines.

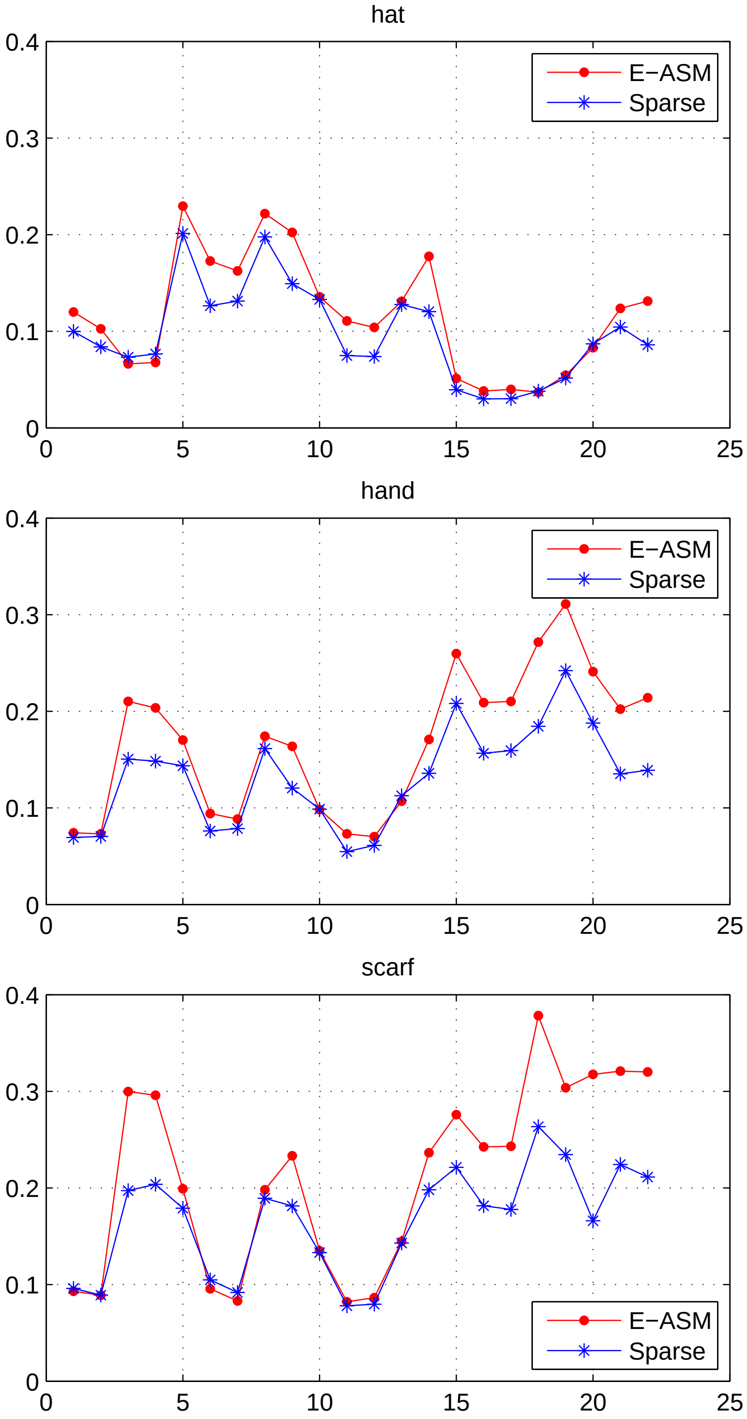

Figure 2.6: Mean localization error on AR database. The proposed method (blue) is significantly better than EASM (red) for the occluded landmarks.

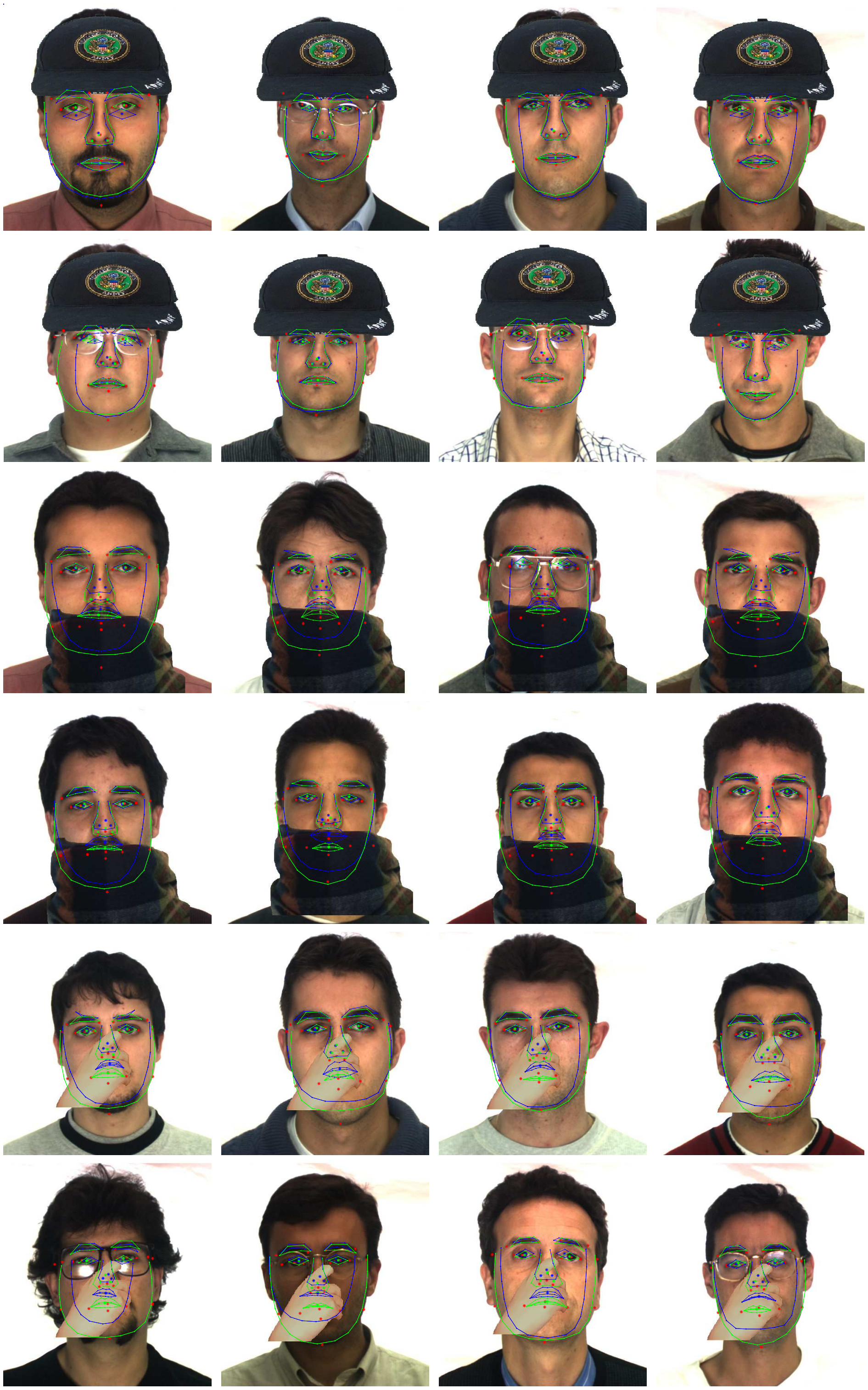

Figure 2.7: Landmark localization results of the proposed method (green) compared with EASM (blue). The ground truth positions are shown in red dots.

2.3 Robust eye localization

Accurate eye localization is a key component of many computer vision systems. Previous research disclosed that poor eye localization significantly degrades the performance of automatic face recognition systems [86]. Despite active research in the last twenty years, accurate eye localization in uncontrolled scenarios remains unsolved. The challenge comes from the fact that the shapes and appearances of eyes change dramatically under various poses and illuminations. Glare and reflections on glasses, occlusions and eye blinks further increase the difficulty to this problem.

Previous research for eye localization can be classified into three categories: geometry based approaches, appearance based approaches and context based approaches. The geometry based approaches model the eyes using geometric information. Yuille et al. [114] described eyes with a parameterized template consisting of circles and parabolic sections. By altering the parameter values, the template is deformed to find the best fit to the image. Bai et al. [6] applied radial symmetry transform to determine the eye centers. A recent work of this category is Valenti et al.'s Isophote Curvature method [97].

In the appearance based approaches, eyes are described by various photometric features including gradients [59], projections [122], edges maps [3], etc. Many statistical classification methods have been applied to model eye appearances. For instance, principal component analysis (eigeneyes) [74], support vector machines [21] [22] [49] [96], multilayer perceptrons [53], neural networks [46], and boosting methods [24] [66] [72], etc. Everingham et al. [39] compared several algorithms and found that the simple Bayesian model outperforms a regression-based method and a Boosting-based method.

The context based approaches incorporate the interaction among objects to disambiguate appearance variation. Active shape models (ASM) [27] and active appearance models (AAM) [30] localize facial landmarks (including eye landmarks) by using a global shape constrain. Cristinacce et al. [33] used pairwise reinforcement of feature responses and a final refinement by AAM. Tan et al. [92] built the enhanced pictorial structure model for eye localization.

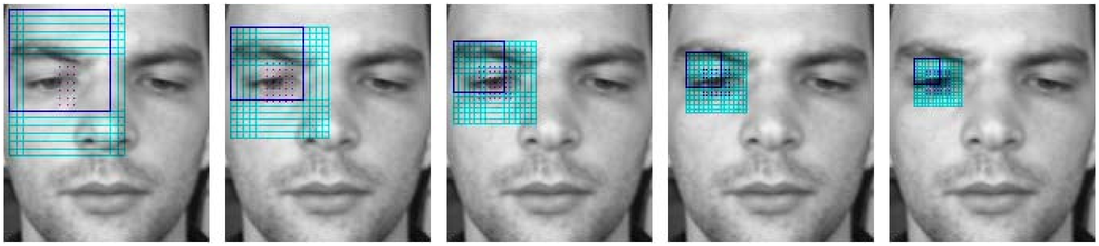

Figure 2.8: Examples of multiscale dictionaries

Although there has been extensive research for eye detection and localization, reliable eye localization in uncontrolled scenarios is still far from being resolved. In uncontrolled scenarios, the geometric structures and eye appearances may be dramatically different from predefined templates or the models learned from the training data. In this case, the accuracy of most previous methods would decrease significantly. In this section, we present a new eye localization method based on Multiscale Sparse Dictionaries (MSD). We built a pyramid of dictionaries that models context information at multiple scales. The localization algorithm starts from the largest scale. At each scale, an image patch is extracted from the previously estimated eye position. We use the sparse dictionary to reconstruct the image patch. The relative location of this patch to the eye is estimated as the position with minimum residual error. The relative location is then used to update the estimations of eye positions.

In our approach, the dictionary of each scale captures the context information of a specific range. Using large context is robust to the variation of eye appearances, and using small context enables more accurate localization. By using context information of multiple scales, our algorithm works both robust and accurately. Our method avoids sliding a search window in the image, thus is more efficient than widely used sliding window methods. Based on sparse representation and optimization methods, the algorithm works efficiently and is resistant to image noises.

2.3.1 Multiscale sparse dictionaries

Recently, the research of computing sparse linear representations with respect to an over complete dictionary has received increasing attention. Sparse methods have been successfully applied to a series of computer vision problems, such as image denoising [38] and face identification [108].

The sparse theory is magnetic as it implies that the signal $x \in R^n$ can be recovered from only $m = O(k \log(n/k))$ measurements [23] if x is a k-sparse signal, which means that x can be well approximated using $k \ll n$ nonzero coefficients under some linear transform. The problem can be formulated with l^0 minimization:

$$x_0 = \arg\min ||x||_0, \quad \text{while } ||y - Ax||^2 < \varepsilon \tag{2.25}$$

where $|| \cdot ||_0$ denotes the l^0-norm which is the number of nonzero entries and ε is the error level. Inspired by the recent work of Wright et al. [108], we first assume that the image patches at the same location do lie on a subspace. Given sufficient training patches $v_{p,i}$ at the location p, we set

$$A_p = [v_{p,1}, v_{p,2}, \cdots, v_{p,N}] \tag{2.26}$$

If a testing patch y is also extracted with the same size from the same location p, it should approximately lie in the linear span of the training patches associated with location p:

$$y = \alpha_{p,1} v_{p,1} + \alpha_{p,2} v_{p,2} + \cdots + \alpha_{p,N} v_{p,N} \tag{2.27}$$

Since the location of the test sample is initially unknown, we define a new matrix A as the concatenation of all the patches extracted from N training images at all locations:

$$\begin{aligned} A &= [A_1, A_2, \cdots, A_P] && (2.28) \\ &= [v_{1,1}, \cdots, v_{1,N}, \cdots, v_{P,1}, \cdots, v_{P,N}] && (2.29) \end{aligned}$$

Then the linear representation of y can be rewritten in terms of all training patches,

$$y = Ax_0 \tag{2.30}$$

where $x_0 = [0, ..., 0, \alpha_{p,1}, \alpha_{p,2} ..., \alpha_{p,N}, 0, .., 0]^T$ is a sparse coefficient vector whose entries

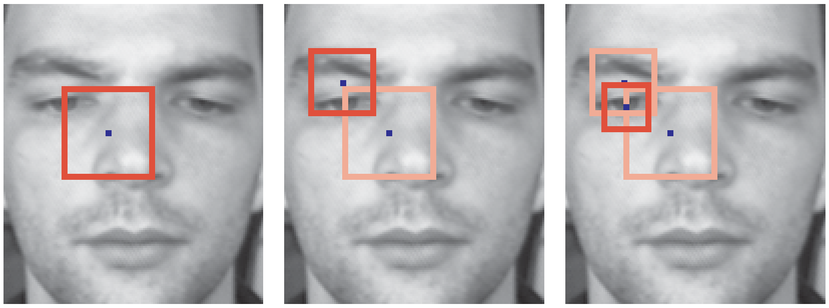

Figure 2.9: Searching eye locations in multiple scales

are zero except those associated with the location p.

The scale of the local patches is an essential factor in sparse representation. Large patches contain more context information, thus are more robust to variation of eye appearances. Small patches contain small context, and are more accurate to localize eye centers. Fig. 2.9 shows how we combine multiscale context information for eye localization. We start from the largest context, which gives an estimate of eye location. Subsequent dictionaries are used in smaller region and are expected to provide a closer eye location. By sequentially applying dictionaries from the largest scale to the smallest scale, the estimated eye location converges to the true position.

To build dictionaries at multiple scales. All training images are carefully aligned using the centers of eyes. The training set are further expanded by rotating and resizing while keeping the eye center fixed. We build dictionaries for left eye and right eye separately. The training patches are extracted by moving a fixed size window around the eye, as shown in Fig. 2.8. At each position, a patch is extracted from the ith image, and forms a column vector $v_{p,i}$. All these vectors are concatenated as A_p in Equation 2.26.

The dictionary A_p can be further compressed via K-SVD algorithm [2], which is an iterative method that looks for the best dictionary to represent the data samples. The training procedure is summarized in Algorithm 4.

Algorithm 4 *Training sparse dictionaries*

```
Align face images using the positions of two eyes.
Expand training set by scaling and rotation.
for scale $s = 1 : S$ do
    for position $p = 1 : P$ do
        Extract image patches at scale $s$ and location $p$.
            $A_{s,p} = [v_{s,p,1}, v_{s,p,2}, \cdots, v_{s,p,N}]$.
        Normalize columns of $A_{s,p}$ to have unit length.
        Compress $A_{s,p}$ by K-SVD.
    end for
    Concatenate all the dictionaries at size $s$.
        $A_s = [A_{s,1}, A_{s,2}, \cdots, A_{s,P}]$
end for
```

2.3.2 Eye localization by using dictionaries

For a testing image, we first use a face detector to find the bounding box of the face. The face region is then cropped and normalized to be the same size of the training faces. The pixels are concatenated into a vector y, which is normalized to have unit length.

The average eye localization of the training faces $L_0 = [x_0, y_0]^T$ are used as initial estimate of the eye location. The localization procedure starts from the largest scale ($s = 1$). Orthogonal matching pursuit algorithm [95] is applied to solve the l^0-norm problem to find k nonzero coefficients.

$$x = \arg\min ||Ax - y||_2 \tag{2.31}$$

$$\text{subject to } ||x||_0 \leq k \tag{2.32}$$

The residual for each non-zero coefficient is computed as

$$r_i(y) = ||y - Ax_i||_2, \ \ (i = 1, \cdots, k) \tag{2.33}$$

Algorithm 5 *Eye Localization*

```
Detect and crop face region.
Set initial eye position $L_0$.
for $s = 1 : S$ do
    Apply OMP algorithm to find $k$ sparse coefficients
        $j_1, \cdots, j_k$
    Find minimum residual and estimate the location of current patch following Equation (9) and (10).
    Update current estimated eye position $L_s$ following Equation (11).
end for
Repeat the above steps for the other eye.
```

The position of current image patch is estimated as the one corresponding to the minimum residual

$$L_y = \arg \min_{\mathrm{i}} r_i(y) \tag{2.34}$$

For image patch y, the previous estimate for its position is L_{s-1} and the new estimate is L_y. We can update the estimate for eye position as

$$L_s = L_{s-1} + L_0 - L_y \tag{2.35}$$

A new image patch is then extracted at scale $s+1$ from location L_s. The previous steps are repeated for each scale. Our localization algorithm is summarized in Algorithm 5.

As shown in Fig. 2.10, we have an estimated location at each scale. By applying the kernel density estimation and mean shift algorithm [26], the final estimated eye location is the position that maximize the following density function

$$f(L) = \frac{1}{S} \sum_{i=1}^{S} K \left(\frac{L - L_i}{scale_i} \right) \tag{2.36}$$

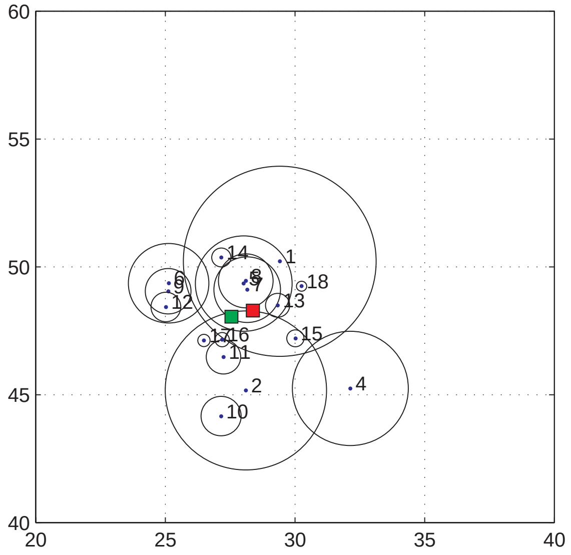

Figure 2.10: Kernel density estimation to estimate eye location

2.3.3 Experiment

In order to assess the precision of eye localization, we apply the normalized error measure introduced by Jesorsky et al. [53]. The normalized error is measured by the maximum of the distances d_l and d_r between the true eye centers C_l, C_r, normalized by the distance between the expected eye centers. This metric is independent of scale of the face and image size:

$$d_{eye} = \frac{max(d_l, d_r)}{||C_l - C_r||} \tag{2.37}$$

We test the precision of our algorithm in the BioID face database [53]. The BioID database consists of 1521 frontal face images of 23 subjects. The images are taken under various lighting conditions in complex backgrounds. Thus this database is considered one of the most difficult databases for eye detection tasks.

We run two-fold cross validation and compare our results with previous methods which report the normalized errors in the same database. The methods we compare with include those used by Jesorsky et al. [53], Hamouz et al. [48] [49], Cristinacce et al. [32], Asteriadis et al. [3], Bai et al.[6], Niu et al. [72], Campadelli et al. [21] [22] and Valenti et al. [97]. For those are inexplicitly reported by the authors, the results are estimated from the graphs in paper.

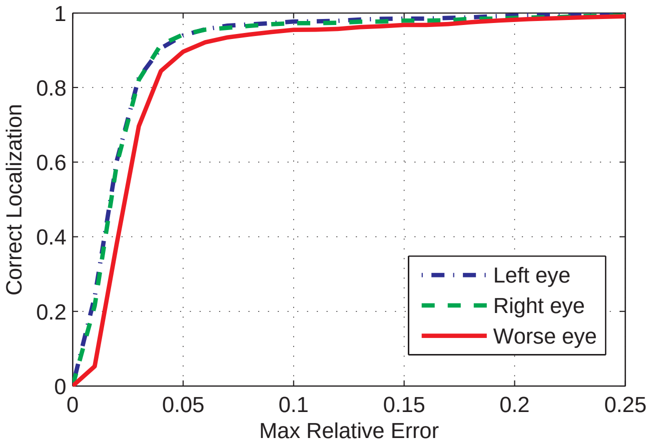

Figure 2.11: ROC curve of eye detection in BioID database

BioID	$e < 0.05$	$e < 0.10$	$e < 0.25$
Jesorsky 01 [53]	40.00%	79.00%	91.80%
Hamouz 04 [48]	50.00%	66.00%	70.00%
Hamouz 05 [49]	59.00%	77.00%	93.00%
Cristinacce 04 [32]	56.00%	**96.00%**	98.00%
Asterialdis 06 [3]	74.00%	81.70%	97.40%
Bai 06 [6]	37.00%	64.00%	96.00%
Niu 06 [72]	78.00%	93.00%	95.00%
Campadelli 06 [21]	62.00%	85.20%	96.10%
Campadelli 09 [22]	80.70%	93.20%	95.30%
Valenti 08 [97]	**84.10%**	90.85%	**98.49%**
Ours	**89.60%**	**95.50%**	**99.10%**

Table 2.1: Comparison of eye localization methods in BioID database

Table 2 compares our results with previous methods for an allowed normalized error of 0.05, 0.1 and 0.25 respectively. Our results and previous best reported results are highlighted in bold text. Specifically, for an allowed normalized error at 0.05 and 0.25, our eye localization algorithm outperforms all previous reported results. And for an allowed normalized error at 0.10, our result is close to the best reported.

2.4 Conclusion

In this chapter, we first review the existing work of facial feature alignment and face tracking. We propose a sparsity driven shape registration method, to handle partial

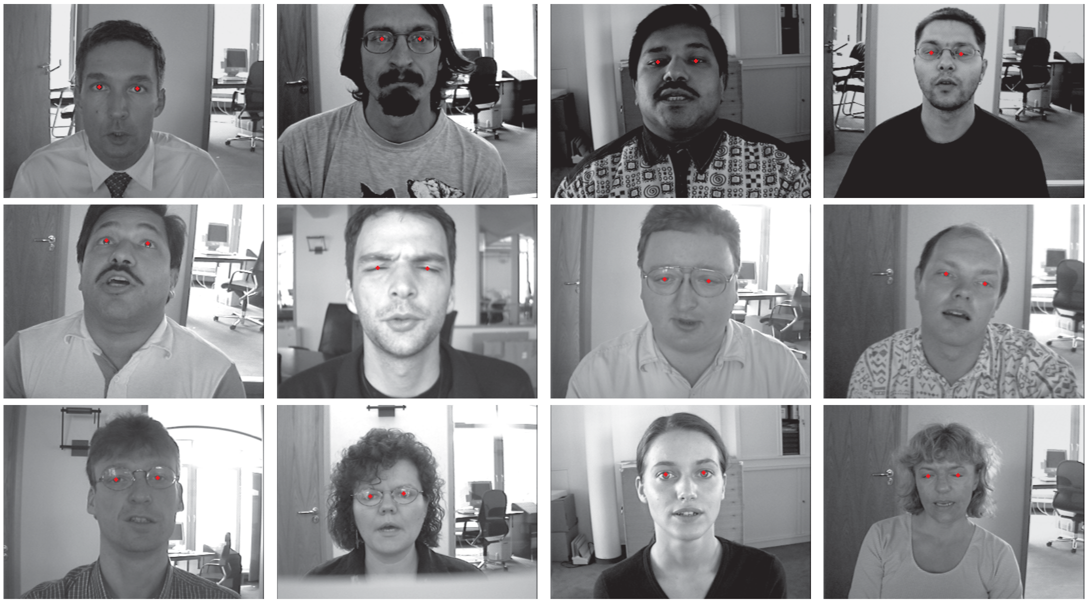

Figure 2.12: Examples of our localization results

face occlusions during registration. By introducing a sparse error term into the linear shape model, our algorithm is more robust for feature localization, especially for the occluded landmarks. Extensive experiments in our synthesized face occlusion database prove the advantage of our method.

We also propose a new method for localizing eye centers. We address the eye localization problem as a sparse coding problem. By assuming that an testing image patch is a linear combination of the training patches at the same position, we propose a new eye localization method by solving sparse coefficients of an over complete dictionary. In the proposed method, we build multiple dictionaries to model context of eyes at multiple scales. Eye locations are estimated from large to small scales. By using context information, our method is robust to various eye appearances. The method also works efficiently since it avoids sliding a search window in the image during localization. The experiments in BioID database prove the effectiveness of our method.

Chapter 3

Facial Component Transfer

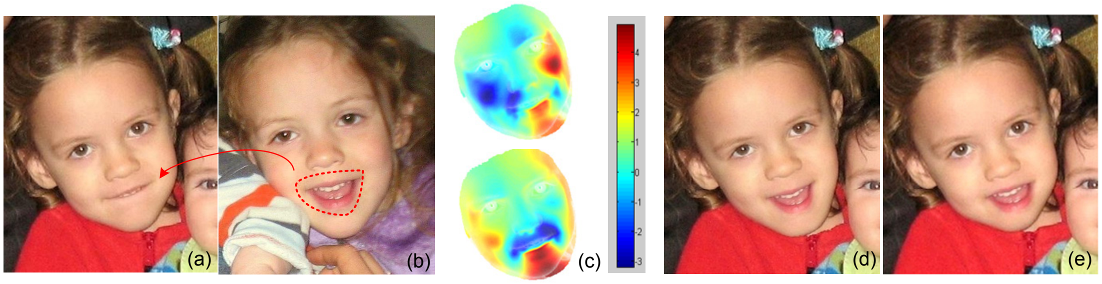

Figure 3.1: Example of applying the proposed expression flow for face component transfer. (a) and (b) are input images, and the user wants to replace the closed mouth in (a) with the open mouth in (b). (c). Expression flow generated by our system, which warps the entire face in (a) to accommodate the new mouth shape. Top: horizontal flow field, bottom: vertical flow filed. (d) Final composite generated by our system. (e). Composite generated using 2D alignment and blending. Note the unnaturally short distance between the mouth and the chin.

Local component transfer between face images with different expressions is a very challenging task. It is well known in the facial expression literature [40] that expressions of emotion engage both signal-intensive areas of the face: the eye region, and the mouth region. For an expression of emotion to appear genuine, both areas need to show a visible and coordinated pattern of activity. This is particularly true of the sincere smile, which in its broad form alters almost all of the facial topography from the lower eyelid downwards to the bottom margin of the face. While general image compositing tools [1] allow the user to crop a face region and seamlessly blend it into another face, they are incapable of improving the compatibility of the copied component and the target face, as the example shown in Figure 5.1. To replace the closed mouth in Figure 5.1a with an open one in Figure 5.1b, a straightforward solution is to crop the mouth region, apply additional alignment adjustments, and seamlessly blend it into the target face. However, the resulting composite is semantically very unnatural (Figure 5.1e). This is because, when the mouth opens, the shape of the whole lower-half of the face deforms accordingly. To our best knowledge there are no existing tools that automatically handle these deformations for creating realistic facial composites.

We address this problem by presenting *Expression Flow*, a 2D flow field applied on the target image to deform the face in such a way that it becomes compatible with the facial component to be copied over. To compute the expression flow we first reconstruct a 3D face shape for each image using a dataset of other people's face shapes. Unlike traditional 3D fitting which tries to minimize the fitting error on each image, we *jointly* reconstruct a pair of 3D shapes, which have the same identity, but with different expressions that match our input image pair. This is formulated as an optimization problem with the objective to minimize the fitting errors with a person identity constraint. A 3D flow is then computed from the pair of aligned 3D shapes, and projected to 2D to form the 2D expression flow. The shapes are also used to warp the 3D pose of the new component before blending in. Due to the identity constraint, the expression flow reflects changes mainly due to differences of expression, and can deform the face in a natural way, as shown in Figure 5.1.

Our expression flow is a hybrid of 3D and 2D methods. On the one hand, we rely on rough 3D shapes to compute the expression difference between faces with different poses. Since typical expression flows contain much lower level of detail (frequencies) than typical appearance details, we found that our rough 3D reconstruction is adequate for the purpose of expression transfer. On the other hand, we rely on 2D methods to warp face images and transfer local details between them. Our system thus has a greater flexibility and a wider application range than previous 3D and 2D expression transfer methods (see Section 3.1).

Based on the proposed expression flow we develop an efficient face compositing tool. To evaluate the effectiveness and generality of the proposed system, we conducted a comprehensive user study. The results suggest that the face composites created by our system have much higher fidelity than those generated by previous methods.

3.1 Related work

Our work is related to previous research on face editing, facial expression mapping, 3D shape fitting and image compositing.

Face Image Editing. Face image enhancement has been the subject of extensive work. Earlier approaches use generic face images as training data for applications such as super-resolution [63] and attractiveness enhancement by global face warping [61]. Recently Joshi et al. [54] proposed a system to adjust global attributes such as tone, sharpness and lighting of a face image using personal priors. Blanz et al. [16] fitted a morphable 3D model to a face image, and then rendered a new face using the same pose and illumination to replace it. The face swapping system [12] achieves a similar goal by constructing and using a large face image library. A real-time system for retrieving and replacing a face photo based on expression and pose similarity was shown in [57]. All these systems target global face editing. However replacing an entire head or face is often not desired for personal photo editing, global warping does not handle large topology and appearance changes, and generating realistic textured head models and compositing them into existing photos remains a challenging problem. Our method combines global warping and local compositing of face parts for an effective by-example expression editing.

Expression Mapping. There is also a large body of works on transferring expressions between images, which falls into two categories: 3D methods and 2D approaches. 3D approaches, such as the expression synthesis system proposed by Pighin et al. [76] and the face reanimating system proposed by Blanz et al. [15], try to create photorealistic textured 3D facial models from photographs or video. 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. These systems thus can only work with high resolution face images shot in controlled indoor environments, and unlike our system, are not robust enough to be used on day-to-day personal face photos.

2D expression mapping methods [105] extract facial features from two images with different expressions, compute the feature difference vectors and use them to guide image warping. Liu et al. [64] proposed 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 view points. Most importantly, these methods alone cannot synthesize features that are not in the original image, such as opening a mouth.

3D Shape Fitting. Recovering the 3D face shape from a single image is a key component in many 3D-based face processing systems. Blanz and Vetter [17] optimized the parameters of a 3D morphable model by gradient descent in order to render an image that is as close as possible to the input image. Romdhani and Vetter [79] extended the inverse compositional image alignment algorithm to 3D morphable models. Shape-from-shading approaches are also applied to 3D face reconstruction [37, 56]. Kemelmacher-Shlizerman et al. [57] showed how to find similarities in expression under different poses, and used a 3D-aware warping of facial features to compensate for pose differences.

Image Compositing. General image compositing tools such as the photomontage system [1] and the instant cloning system [42] allow image regions from multiple sources to be seamlessly blended together, either by Poisson blending [75] or using barycentric coordinates. Sunkavalli et al. [90] proposed a harmonization technique which allows more natural composites to be created.

3.2 Our system

Figure 3.2 shows the flow chart of the proposed system. Given a target face image which the user wants to improve, and a reference image which contains the desired feature to be copied over, our system first uses computer vision techniques to automatically extract facial feature points on both images. Based on the extracted feature points, we then jointly reconstruct 3D face shapes for both images using a 3D face expression dataset. Our 3D fitting algorithm makes sure that the two shapes have the same identity, thus the main difference between them is due to changes in expression. We then compute

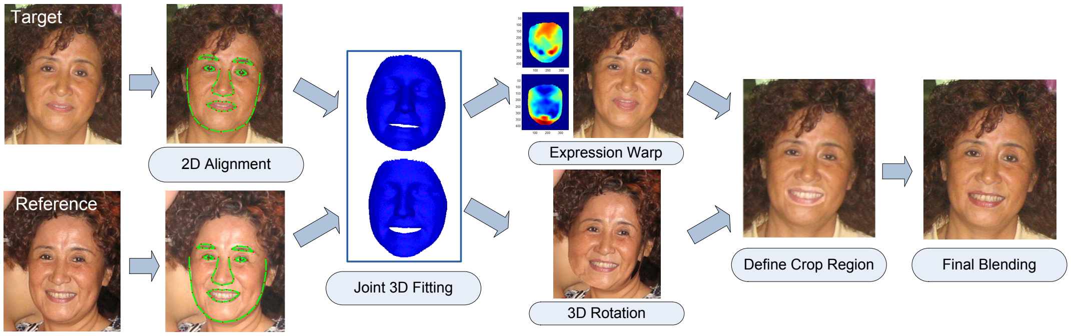

Figure 3.2: The flow chart of the proposed face editing system.

a 3D flow by subtracting the two shapes and project it to 2D to create the expression flow. The expression flow is used to warp the target face. We also use the 3D shapes to align in 3D the reference face to the target face. The user then specifies the region of the facial feature to be transferred, which is then seamlessly blended into the target image to create the final composite.

3.2.1 Single image fitting

We first describe how to fit a 3D face shape to a single face image. Given the input image, the facial landmarks are first localized using Active Shape Model (ASM) [27], a robust facial feature localization method. Following Milborrow and Nicolls's approach [71], we localize 68 feature points, as shown in Figure 3.2.

We represent the 3D geometry of a face with a shape vector

$$s = (x_1, y_1, z_1, \cdots, x_n, y_n, z_n)^T \tag{3.1}$$

that contains X, Y, Z coordinates of its n vertices. Following Blanz and Vetter's work [17], we define a morphable face model using Principal Component Analysis (PCA) on the training dataset. Denote the eigenvectors as v_i, eigenvalues as λ_i, and the mean shape as $\bar{s}$, a new shape can be generated from the PCA model as:

$$s_{new} = \bar{s} + \sum \beta_i v_i = \bar{s} + \mathbf{V} \cdot \beta. \tag{3.2}$$

The 3D fitting is performed by varying the coefficients β in order to minimize the error between the projections of the pre-defined landmarks on the 3D face geometry, and the 2D feature points detected by ASM. We apply a weak perspective projection model, and define the fitting energy for the kth landmark as:

$$E_k = \frac{1}{2}||R \cdot (\bar{s}^{(k)} + \mathbf{V}^{(k)} \cdot \beta) - X^{(k)}||^2,$$

where R is the 2 by 3 projection matrix, $\mathbf{V}^{(k)}$ is the sub-matrix of $\mathbf{V}$ consisting of the three rows that corresponding to X, Y, Z coordinates of the kth landmark. $X^{(k)} = (x^{(k)}, y^{(k)})^T$ is X, Y coordinates of the kth landmark detected from the face image.

Assuming a Gaussian distribution of the training data, the probability for coefficients β is given by:

$$p(\beta) \sim exp[-\frac{1}{2}\sum(\beta_i/\lambda_i)^2]. \tag{3.3}$$

Let $\Lambda = diag(\lambda_1^2, \lambda_2^2, \cdots, \lambda_L^2)$. We define the energy of coefficients as:

$$E_{coef} = \frac{1}{2} \cdot \beta^T \Lambda^{-1} \beta. \tag{3.4}$$

The total energy function to be minimized is thus the combination of the two terms:

$$E = \sum w_k E_k + c \cdot E_{coef}, \tag{3.5}$$

where c is a parameter controlling the tradeoff between the fitting accuracy and the shape fidelity, which is set to 5×10^6 in our system. w_k is the weight for the kth landmark. In our system we set $w_k = 0.5$ for landmarks of eyebrows, since our training shapes are textureless and these landmarks are hard to be labeled accurately. We empirically set $w_k = 2$ for contour points, $w_k = 3$ for mouth points, and $w_k = 1$ for all other points.

To minimize E, we set $\nabla_\beta E = 0$, which leads to:

$$\beta = P^{-1}Q, \tag{3.6}$$

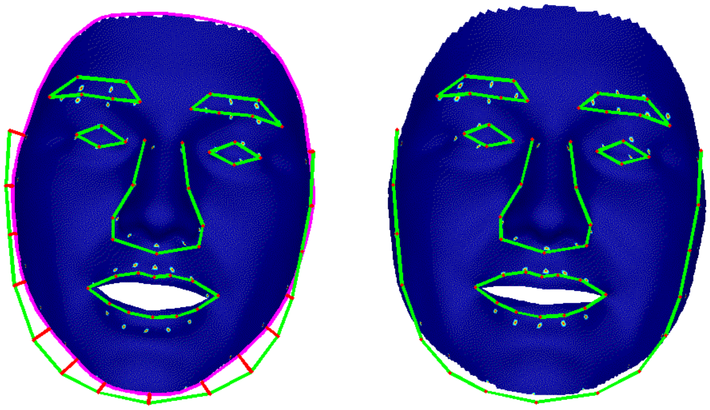

Figure 3.3: Fitting a 3D shape to the target image in Figure 3.2 using our two-stage optimization algorithm. Left: How the shape deforms. Green lines are ASM features lines, the pink line is the projected face contour from face geometry. The short red lines show the contour landmarks projected onto the face contour. Right: fitted face shape after 3 iterations.

where

$$
\begin{aligned}
P &= \sum w_k (R\mathbf{V}^{(k)})^T R\mathbf{V}^{(k)} + c\Lambda^{-1}, && (3.7)\\
Q &= \sum w_k (R\mathbf{V}^{(k)})^T (X^{(k)} - R\bar{s}^{(k)}). && (3.8)
\end{aligned}
$$

The above closed-form solution assumes that we know $V^{(k)}$, the 3D vertices corresponding to the k-th landmark. 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 single corresponding vertex; they must be matched with 3D vertices along the face silhouette. We therefore employ a two-stage optimization approach to find the optimal β. In the first stage we find the correspondences between vertices and landmarks by projecting the vertices onto the image plane, finding their convex hull and assigning each landmark to the closest point on the convex hull, as shown on Figure 3(left). In the second stage we deform the face shape by minimizing the energy in Equation 3.5. We repeat the two stages until the shape converges. Figure 3(right) shows the result after three iterations. We can see that the proposed approach minimizes the fitting error. The algorithm is formally described in Algorithm 6.

Algorithm 6 *Single Image Fitting*

```
Input: facial landmarks $X^{(1),\cdots,(K)}$ and the shape PCA model.
Output: shape $s$ that best fits the landmarks.
Set $\beta = 0$.
repeat
    Set $s = \bar{s} + \mathbf{V}\beta$.
    Find projection matrix $R$ from $s$ and $X^{(1),\cdots,(K)}$ by using the least squares method.
    Project all vertices of $s$ onto the image plane: $s' = P(R, s)$.
    Find the convex hull of $s'$ as $H(s')$.
    For contour landmarks $X^i$, find correspondence using $H(s')$.
    Solve $\beta$ in Equation 3.6.
until $\beta$ converges.
```

3.2.2 Expression models and joint fitting

To train the PCA model we use the face expression dataset proposed by Vlasic et al. [99]. This dataset contains 16 subjects, each performing 5 visemes in 5 different expressions. This dataset is pre-aligned so that the shapes have vertex-to-vertex correspondence.

Building a single PCA model using all training shapes is problematic, since the training shapes vary in both identity and expression. A single PCA might not be expressive enough to capture both types of variations (underfitting), and also does not allow to distinguish between the two. We thus build a PCA model for each expression separately. We could also use more sophisticated nonlinear methods (e.g. manifold [101]). However, since that face shapes do not vary dramatically, we have found that this approximation gives desired results.

For a given image, we select the PCA model that gives the minimum reconstruction error using the fitting algorithm described above. The target and reference face therefore may fall into different expression models. We denote the PCA model for the target image as $(\mathbf{V}^t, \Lambda^t, \bar{s}^t)$, and its training shapes as $\mathbf{S}^t = (s^t_1, ..., s^t_M)$. Similarly, we denote the model and its training shapes for the reference image as $(\mathbf{V}^r, \Lambda^r, \bar{s}^r, \mathbf{S}^r)$. The new shapes to be reconstructed from the images are denoted as s^t and s^r.

Using the constructed expression models and the single image fitting approach proposed above, a natural idea is to fit each input image individually, and then try to generate the expression flow by subtracting the two shapes. However, we found that this approach does not work well. The reason is that each 3D shape is a linear combination of all face shapes in the training dataset, which contains faces from multiple human subjects. By fitting the 3D shape individually to each image, we essentially generate 3D shapes that have different virtual identities. The difference between the two shapes is then mainly due to identity difference, not expression difference.

To solve this problem, our approach is to jointly fit 3D shapes to input images so that they have the same identity. To add such a constraint, we re-formulate s^t as a linear combination of the original training shape vectors s_i^t, parameterized with new coefficients γ_i^t $(i = 1, ..., M)$ as:

$$s^t = \bar{s}^t + \mathbf{V}^t \beta^t = \bar{s}^t + \mathbf{S}^t \gamma^t. \tag{3.9}$$

Similarly, we re-formulate s^r as:

$$s^r = \bar{s}^r + \mathbf{V}^r \beta^r = \bar{s}^r + \mathbf{S}^r \gamma^r. \tag{3.10}$$

The coefficients γ^t and γ^r describe the face shape of a new person under a certain expression as a linear combination of the shapes of the training subjects under the same expression. They essentially define the virtual identities of the two 3D shapes as a linear combination of training subjects. Since s_i^t and s_i^r correspond to the same human subject, by enforcing $\gamma^t = \gamma^r = \gamma$, we guarantee that s^t and s^r have the same identity. We thus replace γ^t and γ^r with a single γ.

From Equation 3.9 we have $\beta^t = (\mathbf{V}^t)^T \mathbf{S}^t \cdot \gamma$. Substituting β with γ in Equation 3.4, the coefficient energy for s^t becomes:

$$E_{coef}^t = \frac{1}{2} \cdot \gamma^T ((\mathbf{V}^t)^T \mathbf{S}^t)^T (\Lambda^t)^{-1} ((\mathbf{V}^t)^T \mathbf{S}^t) \gamma. \tag{3.11}$$

Replacing t with r we have the formulation for the coefficient energy E_{coef}^r for s^r. To

Algorithm 7 *Joint Fitting*

```
Input: facial landmarks of two images, and all PCA models.
Output: shapes $s^t$ and $s^r$ that jointly fit the landmarks on both images.
1: Apply Algorithm 7 to each image to determine their expression models $V^t$ and $V^r$,
   as in Section 3.2.2.
2: Set $\gamma = 0$.
3: repeat
4:     Set $s^t = \bar{s}^t + \mathbf{S}^t\gamma$ and $s^r = \bar{s}^r + \mathbf{S}^r\gamma$.
5:     For each image, apply step 4-7 in Algorithm 1.
6:     Solve the common $\gamma$ in Equation 3.13.
7: until $\gamma$ converges.
```

jointly fit s^t and s^r, we minimize the total energy:

$$E_{total} = \sum w_k(E_k^t + E_k^r) + c \cdot (E_{coef}^t + E_{coef}^r). \tag{3.12}$$

The optimal γ that minimizes this total energy is:

$$\gamma = (P^t + P^r)^{-1}(Q^t + Q^r), \tag{3.13}$$

where P^t and P^r have the same formulation as:

$$P = \sum w_k(R\mathbf{S}^{(k)})^T R\mathbf{V}^{(k)} + c(\mathbf{V}^T\mathbf{S})^T\Lambda^{-1}\mathbf{V}^T\mathbf{S}. \tag{3.14}$$

Substituting $R, \mathbf{S}, \mathbf{V}, \Lambda$ with $R^t, \mathbf{S}^t, \mathbf{V}^t, \Lambda^t$ and $R^r, \mathbf{S}^r, \mathbf{V}^r, \Lambda^r$ gives us P^t and P^r, respectively. Similarly, Q^t and Q^r are defined as:

$$Q = \sum w_k(R\mathbf{S}^{(k)})^T(X^{(k)} - R\bar{s}^{(k)}), \tag{3.15}$$

and substituting $R, \mathbf{S}, X, \bar{s}$ with $R^t, \mathbf{S}^t, X^t, \bar{s}^t$ and $R^r, \mathbf{S}^r, X^r, \bar{s}^r$ gives us the formulation for Q^t and Q^r.

The joint fitting algorithm is formally described as follows:

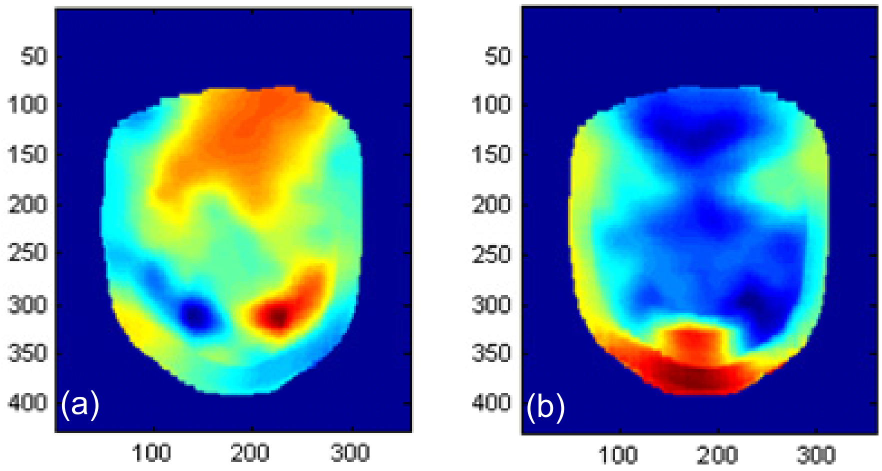

Figure 3.4: 2D expression flow computed from the example shown in Figure 3.2. (a) Horizontal flow field. (b) Vertical flow field.

3.2.3 Computing 2D flow

We first align the two 3D shapes to remove the pose difference. Since the reconstructed 3D shapes have explicit vertex-to-vertex correspondences, we can compute a 3D difference flow between the two aligned 3D shapes and project it onto the image plane to create the 2D expression flow. The flow is further smoothed to remove noise. An example of the final expression flow is shown in Figure 3.4. Figure 3.4a shows the horizontal flow, where red color means positive movement in X direction (to the right), and blue means negative movement (to the left). This figure essentially describes how the mouth gets wider when the person smiles. Figure 3.4b shows the vertical flow, where red color means positive movement along Y axis (moving down), and blue means negative movement (moving up). It illustrates that when the person smiles, her jaw gets lower, and the cheeks are lifted.

As shown in Figure 3.2, by applying the expression flow to the target face, we can warp the target face to have a compatible shape for a larger smile. Similarly, based on the 3D alignment of the two shapes, we can compute a 3D rotation for the reference model, and project it to the image plane to form a 2D alignment flow field, which we call the *alignment flow*. Using the alignment flow, the reference face can be warped to have the same pose as the target face (see Figure 3.2).

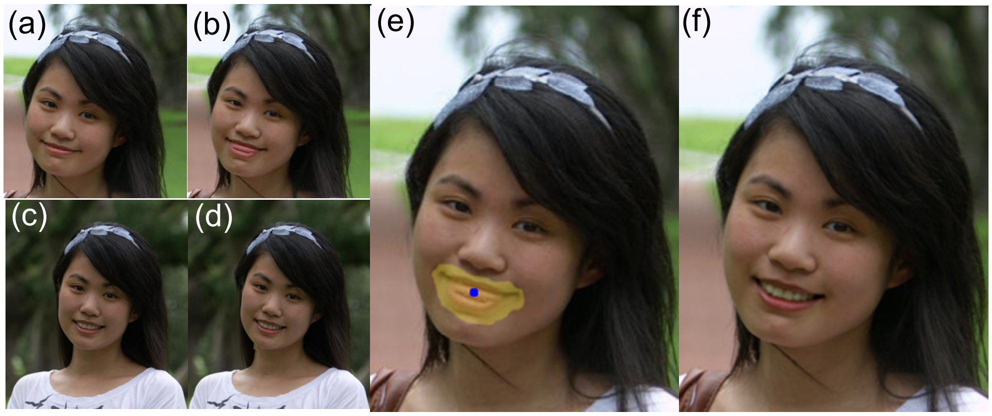

Figure 3.5: Automatic crop region generation. (a) Target image. (b) Warped target. (c). Reference image. (d) Warped reference. (e) The user clicks on the mouth region (marked as blue) to specify the region to be replaced. Our system automatically generates the crop region shown as yellow. (f) Final composite after Poisson blending.

3.2.4 2D compositing

After the two input face images are warped to the desired expression and pose, our system provides a set of editing tools to assist the user to generate a high quality composite. As shown in Figure 3.5, our system employs an interactive feature selection tool, which allows the user to single click a facial feature to generate a crop region that is optimal for blending. This is done by employing a graph cuts image segmentation tool similar to the one proposed in the digital photomontage system [1]. Specifically, the data term in our graph cuts formulation encourages high gradient regions around the user selected pixels to be included in the crop region. For a pixel p, the likelihood for it being included in the crop region is defined as:

$$C(p) = \alpha \exp(-\frac{D_s(p)}{\sigma_d}) + (1-\alpha)\left(1 - \exp(-\frac{\|\nabla S(p)\|}{\sigma_s})\right), \tag{3.16}$$

where $D_s(p)$ is the spatial distance from p to the nearest pixel selected by the user, $\|\nabla S(p)\|$ is the gradient magnitude at p, and σ_d, σ_s and α are parameters controlling the shape and weight of each term. $L(p)$is the label of p. The data penalty in the graph cuts formulation is then defined as $C_d(p, L(p)) = 1 - C(p)$ if $L(p) = 1$ (inside the crop region), and $C_d(p, L(p)) = C(p)$ if $L(p) = 0$ (outside the crop region).

We choose to use the “match gradient” formulation in the photomontage system for setting the neighborhood penalty $C_i(p, q, L(p), L(q))$ as:

$$\|\nabla S_{L(p)}(p) - \nabla S_{L(q)}(p)\| + \|\nabla S_{L(p)}(q) - \nabla S_{L(q)}(q)\|, \tag{3.17}$$

which can lead to fewer blending artifacts. The total energy function which is the sum of the data and neighborhood penalty is then minimized by graph cuts optimization [18].

Once the crop region is computed, we apply additional harmonization steps to make the cropped region more compatible with the target image. The most noticeable artifact we found is that after applying the alignment flow to warp the reference image, it becomes blurry. Blending a blurry region into a sharp image can be very noticeable. To alleviate this problem we first apply the wavelet-based detail enhancement filter proposed in [43] to sharpen the crop region, then blend it into the target image using the Poisson blending method [75].

3.2.5 User assistance

The computer vision components of our system cannot work perfectly well in all cases. For difficult examples, our system requires a small amount of user assistance in order to generate high quality results. The main steps that require user intervention are 2D alignment using ASM and crop region specification.

Our ASM implementation sometimes cannot generate accurate 2D alignment results for side-view faces with large rotation angles. This is a known hard computer vision problem. An example is shown in Figure 3.6a, where some of the automatically computed landmarks are not accurate, especially for the mouth and the left eye region. Using these landmarks for 3D fitting is then erroneous. In our system we allow the user to manually correct the bad ones, so that accurate 3D fitting can be achieved, as shown in Figure 3.6b.

The crop region generation tool described in Section 3.2.4 allows the user to quickly specify a selection mask. However, this method sometimes cannot capture the subtle semantic expression details that the user wants to transfer. Such an example is shown in Figure 3.6d, where the automatically generated crop region misses the unique smile

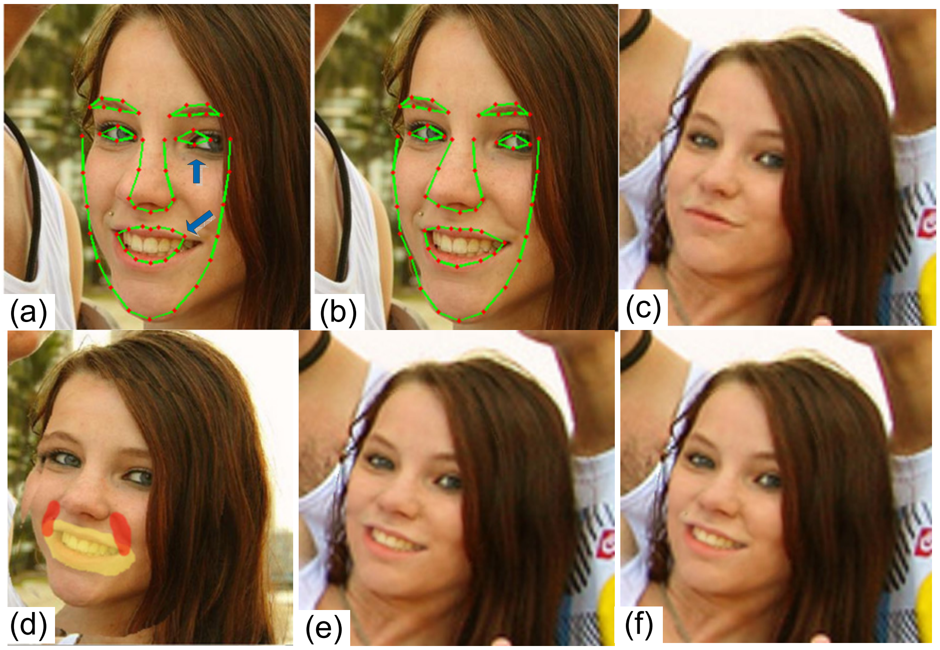

Figure 3.6: User assistance modes. (a) Reference image with automatically extracted landmarks. Errors are highlighted by blue arrows. (b) Landmark locations after manual correction. (c) Target image. (d) Automatically computed crop region (yellow) with user correction (red) to add smile folds. (e) Composite without smile folds. (f) Composite with smile folds.

folds of the subject. The user can manually add the smile folds into the crop region, which leads to a more natural composite shown in Figure 3.6f.

3.3 Results and evaluations

3.3.1 User study

To quantitatively and objectively evaluate our system, we conducted a user study using Amazon Mechanical Turk (AMT). Our evaluation dataset contains 14 examples, each including four images: two originals, the result generated by our system and the result generated by a 2D method. The 2D method first applies Lucas-Kanade image registration [65] between the two faces using only pixels inside the face region, using the detected fiducial points for initialization, and then uses the rest of our system to create the final composite. This is similar to the state-of-art local facial component transfer approaches such as the photomontage system [1] and the face replacement feature in

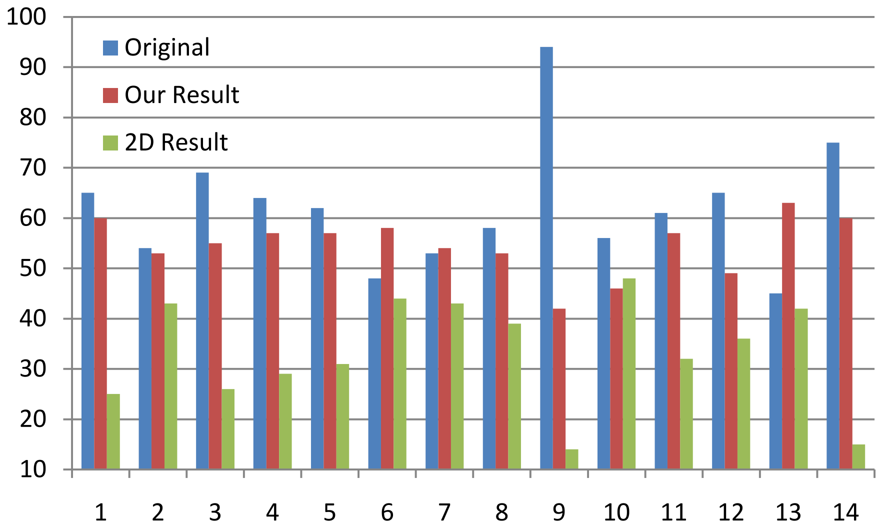

Figure 3.7: User study results on comparing the original images, our results and 2D results. Vertical axis is the number of times that a specific category is voted for by the users.

Photoshop Elements. These examples span across different age groups from small children to elders, as well as different ethnic groups, and include both men and women. For each user and each example, two images out of three (original, our result and 2D result) were randomly chosen to be shown side-by-side, and the user was asked to select the one that appears more natural. Each combination was evaluated by 50 different users, so each result was compared against the originals and the other result both for 50 times. On average the users spent 15 seconds to evaluate each pair.

Figure 3.7 shows the user study results. As we can see, the original images were typically rated as most natural. This is not a surprise as humans are very sensitive to the slightest imperfection on faces, and we do not expect our results to be more realistic than natural face images. Surprisingly however, in example 6, 7 and 13, our results were actually rated higher than the originals. We believe this is because our results in these examples achieved almost the same level of fidelity as the originals, and the users were essentially rating which face has a more pleasant expression when they did not see noticeable artifacts (see example 7 in Figure 3.8).

As the data shows, our method was consistently favored by the users against the

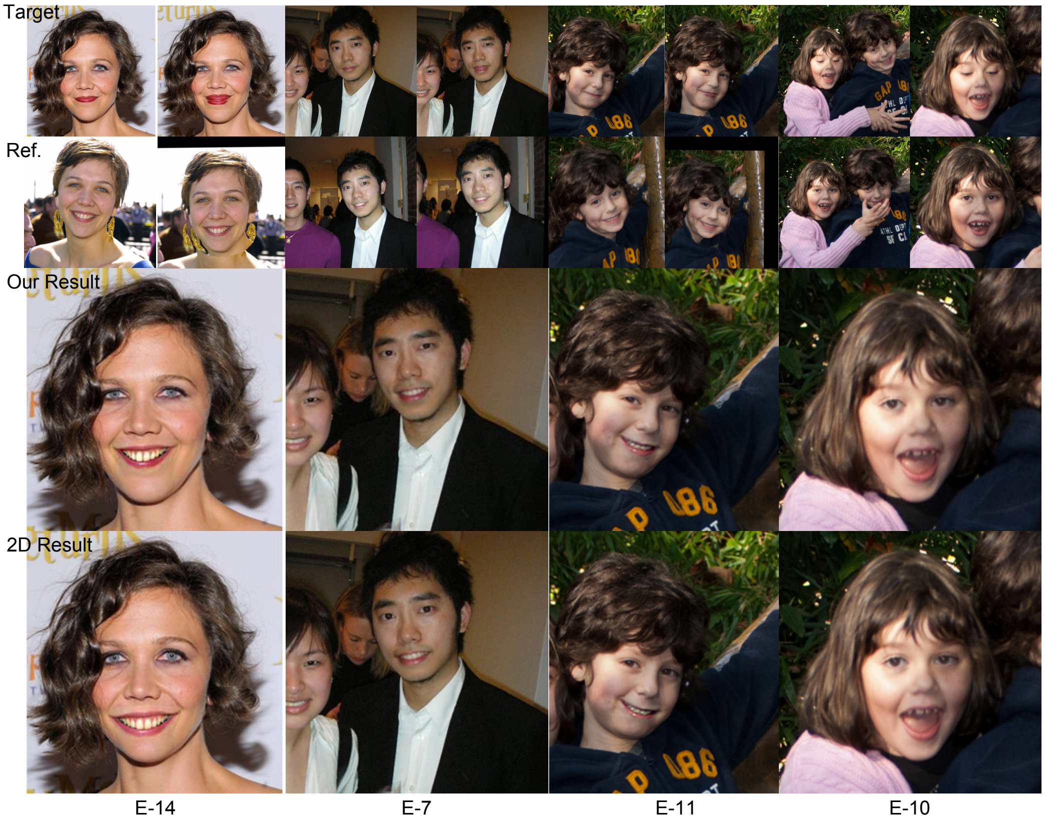

Figure 3.8: Example 14, 7, 11, 10 used in the user study. For each example, top row: target image (left) and after being warped by the expression flow (right); second row: reference image (left) and after being warped by the alignment flow (right); third row: our result; last row: 2D result.

2D results by a significant margin, with the exception of example 10, which is an eye-replacement example (last column in Figure 3.8). This suggests that sometimes the 2D method is sufficient for eye replacement when the two faces have roughly the same pose, since the upper-half of the face is more rigid and opening or closing the eyes may not involve any significant global change to the face. The expression flow is insignificant in this case.

Some examples used in the user study are shown in Figure 5.1, 3.5, 3.6 and 3.8. All other examples are included in the supplementary material. We consider example 9 to be a failure case and will address it in next section.

To further evaluate the effectiveness of the proposed expression flow, we conducted another user study where we only compare our results against those generated by disabling expression flow on the target image. Since 3D alignment is still applied, these

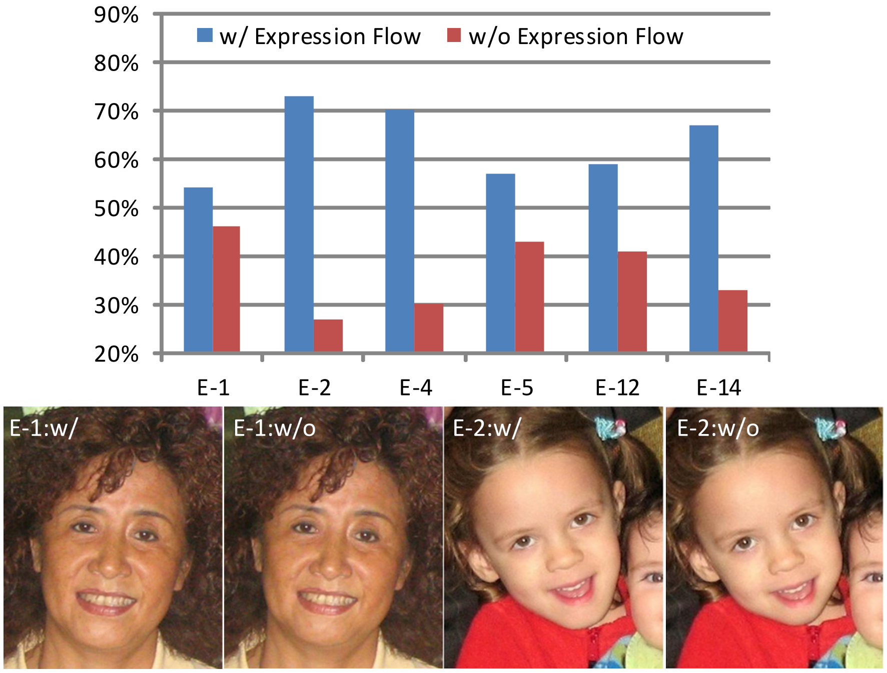

Figure 3.9: Top: user study results on comparing results with and without applying expression flow. Vertical axis is the percentage of users favoring results with expression flow. Bottom: Examples with the most significant (E-2) and insignificant (E-1) differences.

results are more natural than the 2D results. We chose 6 examples on which our method were rated significantly better than 2D method, and conducted a second round side-by-side comparison on AMT. Each pair was evaluated by 100 users. The results are shown in Figure 3.9. This study clearly suggests that the users consistently favored results with expression flow being applied.

3.3.2 Comparison with general face modeller

There are existing general face modellers that can construct a 3D face model from an input image. One may wonder if they can be applied to build 3D models for computing the expression flow, instead of using the 3D fitting method proposed in this section. To test this idea we applied two single image 3D fitting methods, the popular FaceGen Modeller [88] and proposed fitting algorithm, to each face separately, as shown in Figure 3.10. Note that the difference flow computed using single image fitting significantly distorts the faces, and the final composites are much worse than our results shown in

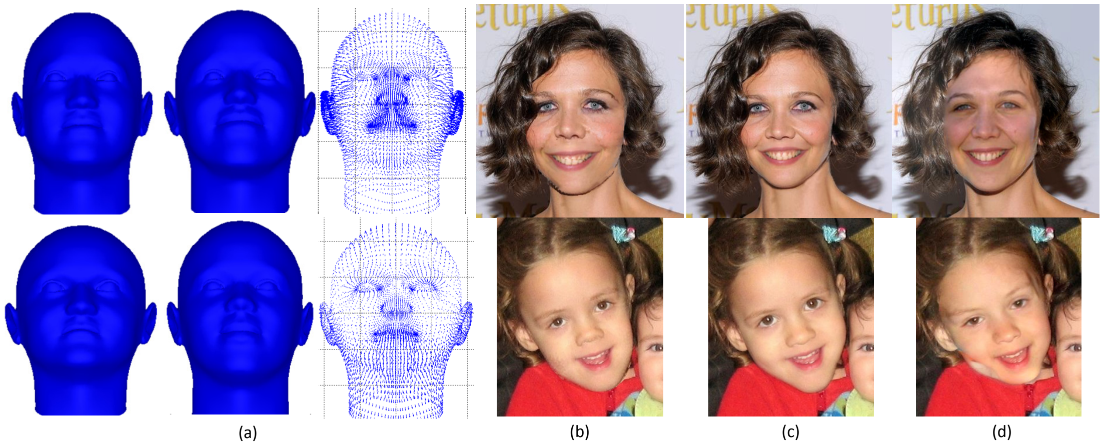

Figure 3.10: Comparisons with other methods. (a). 3D Models and the difference flows generated by FaceGen Modeller. (b). Composites generated using FaceGen models. (c). Composites generated using the single image fitting algorithm. (d). Whole face replacement results. Compared with our results in Figure 5.1 and 3.8, these results contain much more severe artifacts.

Figure 5.1 and 3.8. This is because single image fitting methods will vary all possible internal parameters to best fit a 3D model for a face image, thus the two 3D models contain not only expression difference, but also *identity difference*. Using this difference flow to warp the face will lead to significant artifacts.

In Figure 3.10d we also show comparison results by replacing the whole target face with the 3D-corrected reference face generated by our system, inspired by the face replacement system of [12]. Note the various artifacts around the hair region in both examples as whole face replacing cannot deal with occlusions properly. More importantly, there is inconsistent lighting in the top example and the changed gaze direction in the bottom example. This suggests that whole face compositing is not always reliable nor is it always desirable. Local component transfer is preferable in many cases.

3.3.3 More experiments

Instead of being applied for transferring facial components from one image to another, the expression flow can also be used directly for expression enhancement that does not involve large topology changes, e.g., opening a mouth. Figure 3.11 shows two such examples, where the target face has a neutral or slight smile expression and the reference

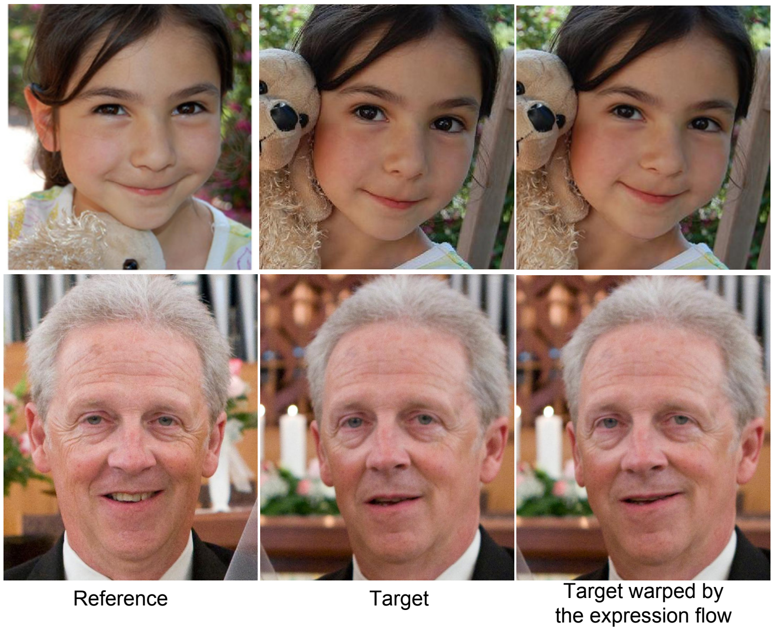

Figure 3.11: Using expression flow for expression transfer only.

face has a large smile. In this case the computed expression flow accurately captures the characteristics of the person's smile, thus can be applied on the target image for smile enhancement. Since no blending is applied, these results have very high fidelity.

Note that previous expression mapping techniques, such as the expression ratio image [64], cannot be directly applied in these cases due to the 3D pose difference between input face images.

In some cases the user may only want to transfer a local component without modifying the other correlated ones. As the example shown in Figure 3.12, one may only want to copy the eyebrows and wrinkles from the reference face to the target face to make the expression more expressive. In this case we can disable the expression flow, and only apply the alignment flow computed from the 3D models to the reference face. Compared with the 2D result, our composite is more realistic since the correct 3D transformation has been applied to the eyebrows and wrinkles.

Although most examples we have shown aim at changing a non-smiling face to a smiling one, our system can handle other less common expressions within the range of expressions in our training data set. Figure 3.13 shows two examples where we change a neutral face to a frown expression. Note how the eyes of the person changes along

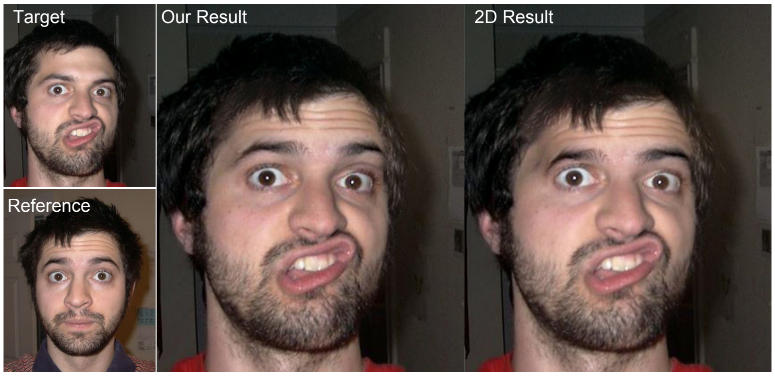

Figure 3.12: An example of creating composite without applying expression flow. Eyebrows and wrinkles are transferred from the reference to the target image. Note the right eyebrow in the result images.

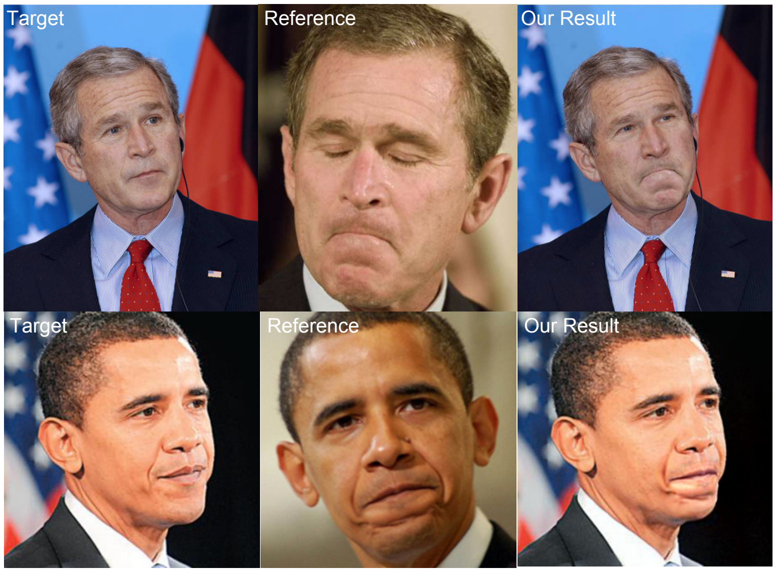

Figure 3.13: Changing a neutral expression to a frown expression.

with the expression in the top example, and the large 3D pose difference between two input images in the bottom example.

We found that the iterative joint fitting algorithm converges fast in practice. In our experiments we use only 10 iterations, each being a closed-form solution. In our Matlab implementation, it takes less than one second to run the algorithm. Our approach is also robust to local minima for two reasons. First, although we use an alternating minimization approach, in each stage we have a closed-form solution to efficiently minimize the energy in that stage. Second, we found that our initialization by aligning the 3D shapes to the images using internal facial landmarks, is accurate enough and leads to a quick convergence in all our examples.

Our system does not always generate realistic composites. In general, if the input faces have a very large pose difference, then the face region cropped from the reference image has to go through a large geometric transformation before being composed onto the target image. Artifacts are likely to be introduced in this process. Furthermore, in difficult cases our 3D fitting may contain errors, which will lead to inaccurate transformation. Figure 3.14 (top) shows Example 9 in the user study in Figure 3.7. Our system failed to compute an accurate transformation for the mouth region, thus in the result the mouth region is clearly not compatible with the pose and shape of the target face, although our result is still significantly better than the 2D result. To avoid this problem one can find another reference image where the face pose is closer to that of the target face. This is not a problem if a large personal photo album of the subject is available.

Figure 3.14(bottom) shows another limitation of our system on handling asymmetric expressions. For this expression transfer example, our system cannot raise one eyebrow and at the same time lower the other one. This is because our training dataset contains only symmetric expressions. Using a richer training dataset will help in this case.

3.4 Conclusion

In this chapter we address the problem of transferring local facial components between face images with different expressions of the same person. To account for the expression difference, we propose a novel expression flow, a 2D flow field which can warp the target face in a natural way so that the warped face becomes compatible with the new component to be blended in. The expression flow is computed from a novel joint 3D fitting method, which jointly reconstructs 3D face shapes from the two input images, so that the identity difference between them is minimized, and only expression difference exists. A comprehensive user study was conducted to demonstrate the effectiveness of our system.

Our system currently uses the face expression dataset collected by Vlasic et al. [99].

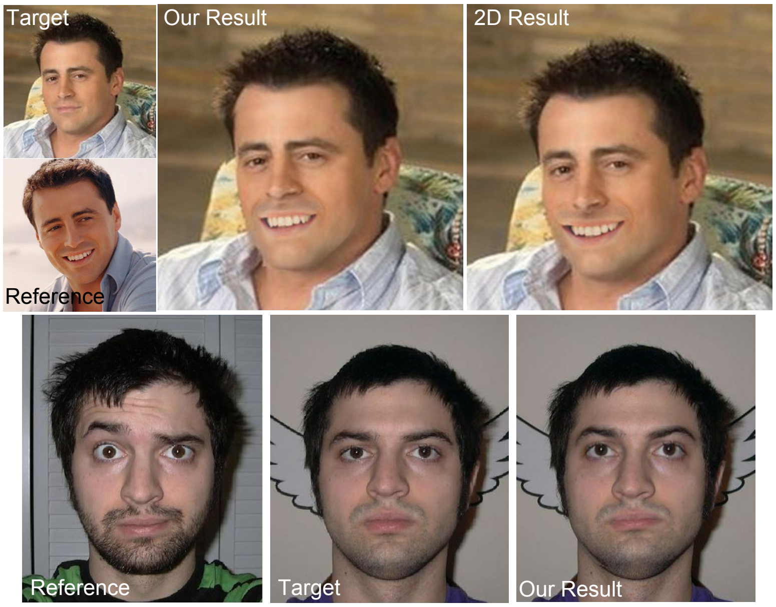

Figure 3.14: Two failure examples. Top: an unsuccessful compositing. Bottom: an unsuccessful expression transfer.

While we demonstrate in this paper that our system can work reliably well on a wide variety of people of different races, ages and genders, we are also aware that the dataset is not rich enough to handle all possible expressions, especially asymmetric ones.

As pointed out by some user study subjects, some of our results still contain minor but noticeable photometric artifacts. For instance, some subjects pointed out that the mouth region shown in Example 14, Figure 3.8 is grainy. To fix these blending artifacts, more advanced harmonization methods [90] can be incorporated into our system to further improve the quality of the final results.

Chapter 4

Facial Expression Editing

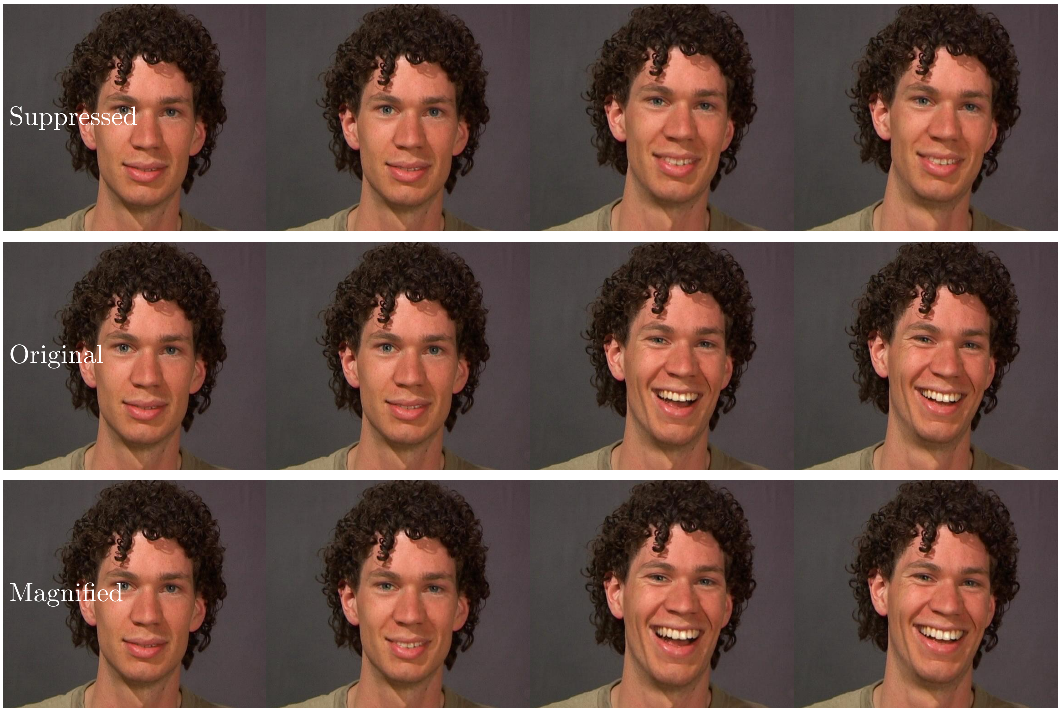

Figure 4.1: We can magnify or suppress an expression in video. **Middle:** Frames from the original video. **Top:** Synthesized frames in which the smile is suppressed. **Bottom:** Synthesized frames in which the smile is magnified.

In this chapter, we focus on semantic-level editing of expressions in video, such as magnifying a smile (Figure 4.1) 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 space. Our method results in high fidelity reconstruction and has some robustness to viewpoint variation.

4.1 Related work

Manipulating and replacing facial expressions in photographs and videos has gained more attention in recent years [104]. 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.* [76] and the face reanimating system proposed by Blanz *et al.* [16]. 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 [105] extract facial features from two images with different expressions, compute the feature difference vectors and use them to guide image warping. Liu *et al.* [64] proposed 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.* [94] applied Active Appearance Models (AAMs) [30] 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 [112]. 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.* [19] to drive a video by audio, and by Kemelmacher *et al.* [58] to generate smooth transitions from a personal photo collection. Kemelmacher-Shlizerman *et al.* [57] 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.* [112] 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 this by 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 [98] 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.* [99] 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.* [34] 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.

4.2 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 Vlasic et al. [99]. 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 [99], we define a morphable face model using multi-linear decomposition, which decomposes the 3D shape into expression and identity:

$$s_t = \bar{s} + V \times_\beta \beta_t \times_\gamma \gamma \tag{4.1}$$

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

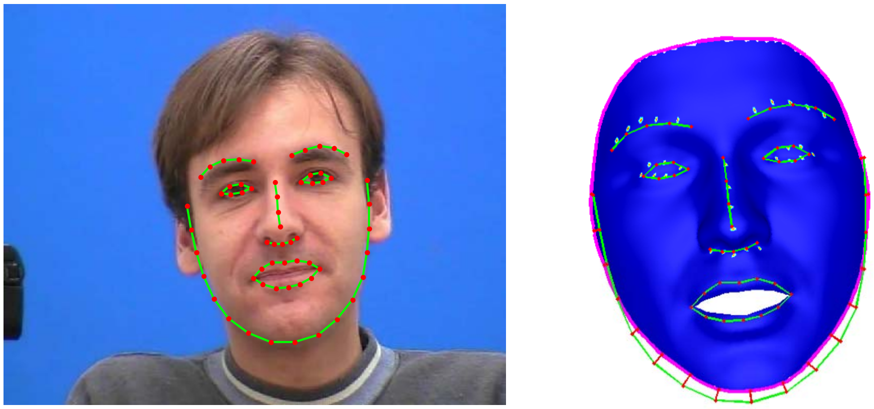

Figure 4.2: **Left:** Facial features detected and tracked by Active Appearance Model (AAM). **Right:** Updating face contour-landmark correspondences. The green curves connect all AAM features and the pink curve is the contour of the projected face geometry. The short red lines show the landmarks projected onto the face contour.

points. The 2D feature points are detected and tracked by using Active Appearance Model (AAM) [30] 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.* [81] to track 66 facial features Y_t for each frame t, as illustrated in Figure 4.2.

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:

$$X_t = \mathbf{R}_t \mathbf{L}_t s_t = P_t s_t \tag{4.2}$$

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:

$$E_f = \sum_t ||W^{1/2}(P_t s_t - Y_t)||^2 \tag{4.3}$$

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:

$$E_\gamma = \frac{1}{2}\gamma^T\gamma \tag{4.4}$$

and:

$$E_\beta = \frac{1}{2}\sum_t \beta_t^T\beta_t \tag{4.5}$$

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,

$$E_e = \frac{1}{2}\sum_t(\lambda_1||\nabla_t\beta_t||^2 + \lambda_2||\nabla_t^2\beta_t||^2) \tag{4.6}$$

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

$$E = E_f + \lambda_\gamma E_\gamma + \lambda_\beta E_\beta + E_e \tag{4.7}$$

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

4.2.1 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

Algorithm 8 *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
```

four steps are iterated until convergence. To initialize, we set all β_t's to zero, and γ to a random vector with unit length.Our algorithm is summarized in Algorithm 8.

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 [50], which assumes that pixels are square and the skew coefficient between x and y is zero, 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 [73] to minimize the geometric error.

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 Figure4.2, 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.

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:

$$\frac{\partial E}{\partial \gamma} = \sum_t M_t^T W (P_t \bar{s} + M_t \gamma - Y_t) + \lambda_\gamma \gamma = 0 \tag{4.8}$$

in which:

$$M_t^{(\gamma)} = P_t (V \times_\beta \beta_t) \tag{4.9}$$

By solving the above equation, we get:

$$\gamma = A^{-1} B \tag{4.10}$$

in which:

$$A = \sum_t A_t + \lambda_1 I \tag{4.11}$$

where:

$$A_t = M_t^T W M_t \tag{4.12}$$

and:

$$B = \sum_t B_t \tag{4.13}$$

where:

$$B_t = M_t^T W (X_t - P_t \bar{s}) \tag{4.14}$$

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:

$$\begin{aligned}\frac{\partial E}{\partial \beta_t} &= M_t^T W(P_t\bar{s} + M_t\beta_t - Y_t) + \lambda_\beta \beta_t \\ &+ \lambda_1(-\beta_{t-1} + 2\beta_t - \beta_{t+1}) \\ &+ \lambda_2(\beta_{t-2} - 4\beta_{t-1} + 6\beta_t - 4\beta_{t+1} + \beta_{t+2}) \\ &= 0 \end{aligned} \tag{4.15}$$

in which:

$$M_t^{(\beta)} = P_t(V \times_\gamma \gamma) \tag{4.16}$$

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:

$$\beta = A^{-1}B \tag{4.17}$$

in which:

$$A = diag(A_t) + \lambda_\beta I + \lambda_1 H_1 + \lambda_2 H_2 \tag{4.18}$$

and:

$$B = [B_1^T, \ldots, B_T^T]^T \tag{4.19}$$

Here A_t and B_t are defined in the same form as in Eqn. 4.12 and Eqn. 4.14, replacing M_t with $M_t^{(\beta)}$. H_1 and H_2 control the temporal smoothness and are defined as:

$$H_1 = (K_1^T K_1) \otimes I_m \tag{4.20}$$

$$H_2 = (K_2^T K_2) \otimes I_m \tag{4.21}$$

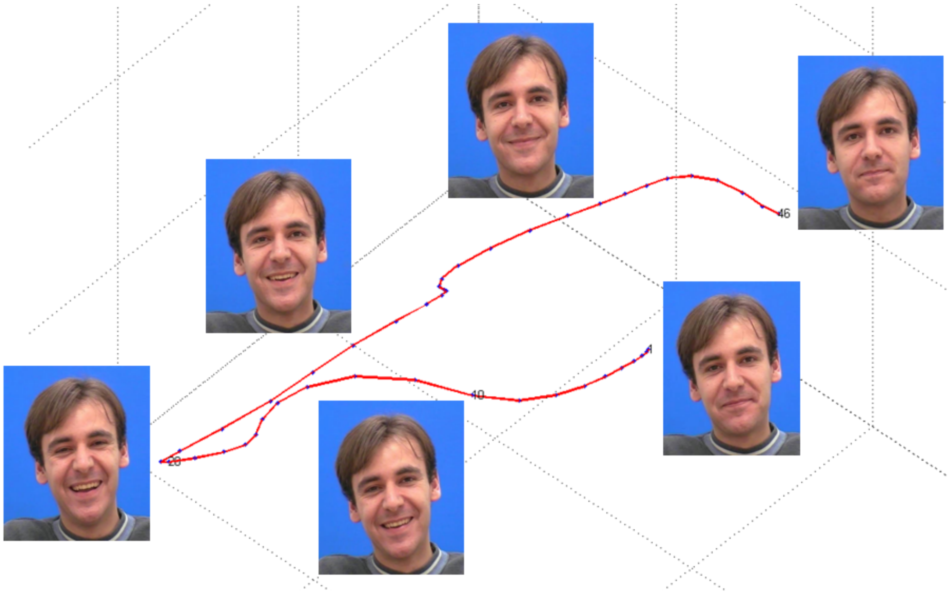

Figure 4.3: As the person goes from neutral to smiling and back to neutral expression, we are able to infer the expression coefficients over time β_t and plot their trajectory in 3D. The video clip is from T. Cootes *et al.* "Talking face video".

and

$$K_1 = \begin{bmatrix} 1 & -1 & \\ & \dots & \\ & -1 & 1 \end{bmatrix}_{(T-1)\times T} \tag{4.22}$$

$$K_2 = \begin{bmatrix} -1 & 2 & -1 & \\ & \dots & & \\ & -1 & 2 & -1 \end{bmatrix}_{(T-2)\times T} \tag{4.23}$$

4.2.2 Fitting a sequence

We evaluate the proposed fitting algorithm using a sequence of 46 frames from the "Talking Face Video" [29]. 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 Figure 4.3. As expected the trajectory starts from the neutral expression point, goes towards smiling and back to neutral.

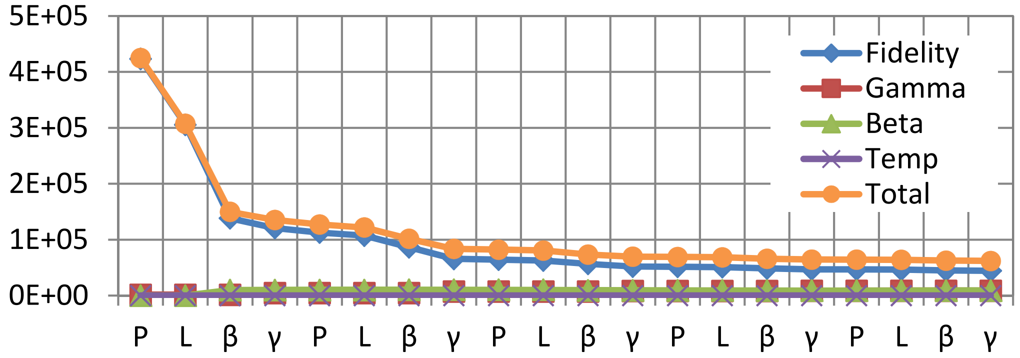

Figure 4.4: The energies after each step in the first five iterations. The total energy decreases monotonically and converges quickly.

We plot the total energy E and the fitting error term E_f, the shape energy E_β and E_γ, and the temporal coherence term E_t after each step in the first five iterations. As shown in Figure 4.4, the total energy monotonically decreases in each step and our method converges.

4.3 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. 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. 4.4).

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" [112][111]. 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

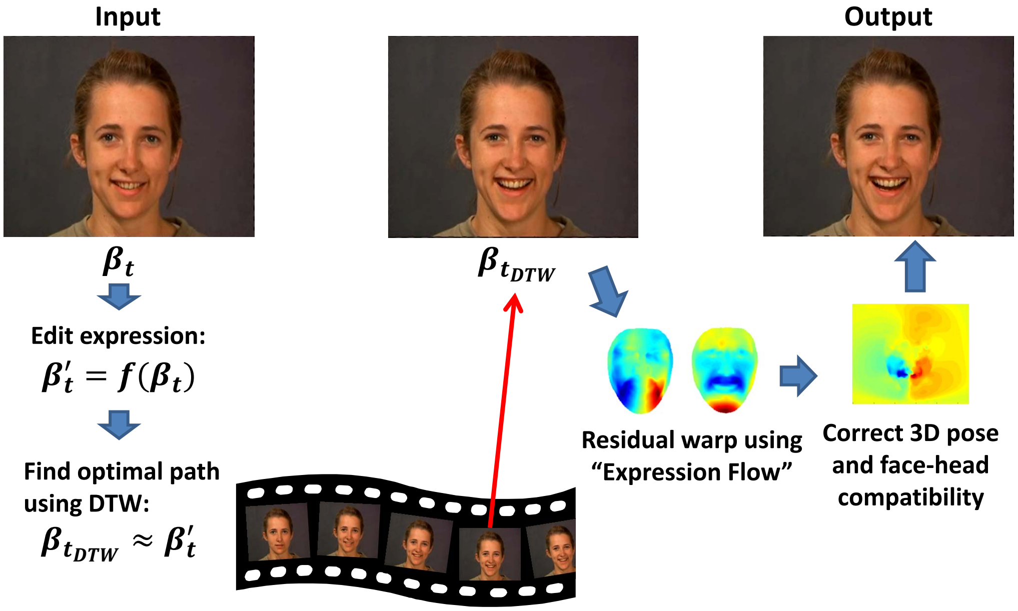

Figure 4.5: Expression manipulation process. For each input frame we define a desired expression manipulation $\beta'_t = f(\beta_t)$. Then we use Dynamic Time Warping to find an optimal sequence of frames from the input video $\beta_{t_{DTW}}$ that is both close to the desired expressions and is temporally 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.

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) [80] 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 Figure 4.5.

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: $D(i,j) = ||\beta'_i - \beta_j||_2$. Figure 4.6 shows an example. In the original video the subject

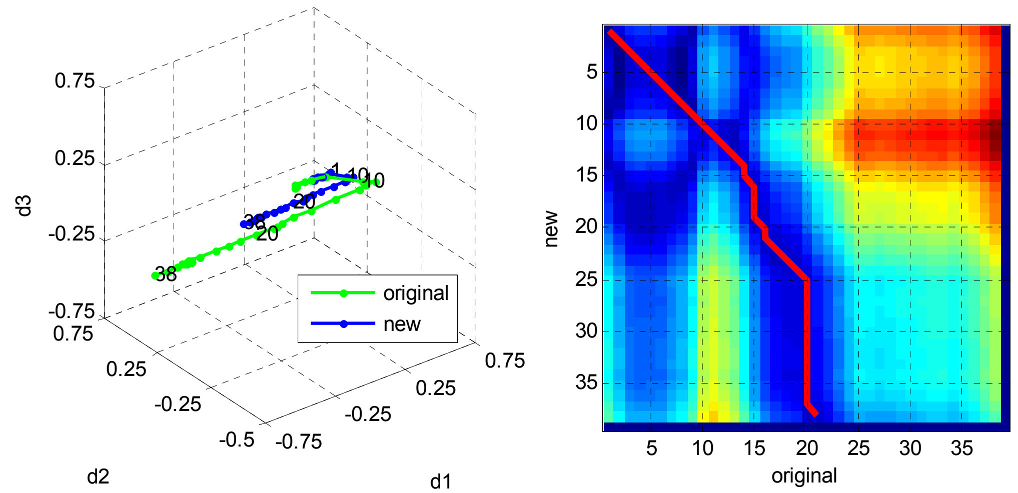

Figure 4.6: Expression neutralization. **Left:** The original expression coefficients β_t (green curve) is scaled by factor 0.5 (blue curve). **Right:** Frame correspondence computed using Dynamic Time Warping (red curve on top of the frame distance matrix).

changes expression from neutral to full smile. The original expression coefficients β_t (green curve on Figure 4.6, left) are scaled by a factor of 0.5 (blue curve) to neutralize the smile. Figure 4.6 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:

$$F_{ij}^{(face)} = R_i V \times_\beta \beta_i' \times_\gamma \gamma - R_j V \times_\beta \beta_j \times_\gamma \gamma \tag{4.24}$$

In our case we use Expression Flow to warp the output of DTW ($\beta_j = \beta_{t_{DTW}}$) to the desired expression ($\beta_i = \beta_t'$). Figure 4.7 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 face boundary move

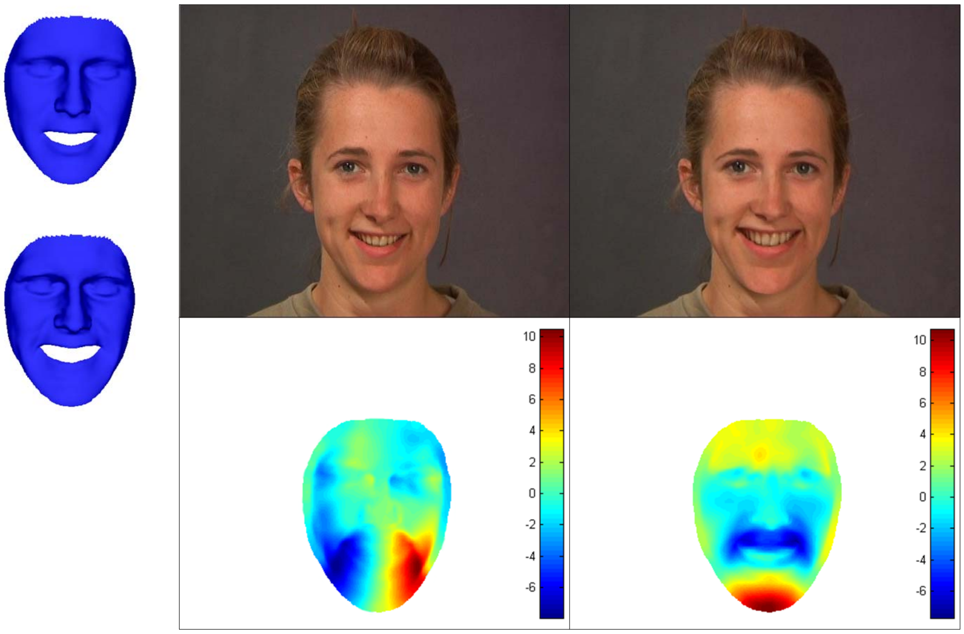

Figure 4.7: Exaggerating the smile using face flow. **Left margin:** The original 3D shape and, below it, the modified one after changing the expression coefficients using equation 4.25. **Top:** Original image (left) and the warped result (right). **Bottom:** The flow in X (left) and Y (right).

the same way. To warp the background we compute the optical flow [62] 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 [84] approach to smoothly blur the difference between the two flows.

4.4 Applications and results

4.4.1 Changing the magnitude of an expression

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

$$\beta'_t = \beta_0 + \alpha(\beta_t - \beta_0) \tag{4.25}$$

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 Figure 4.1 and Figure 4.11.

4.4.2 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 Figure 4.8.

4.4.3 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 Figure 4.9. 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.

4.4.4 Discussion

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 (i.e. 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 [56] to our similarity and improve the compositing method [112] to alleviate this problem. Second, when magnifying an expression, there is a limit to how much

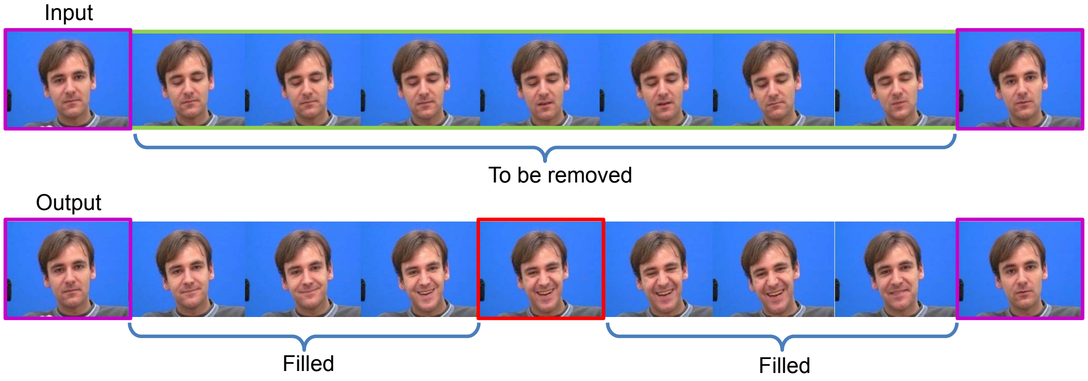

Figure 4.8: Replacing expression. **Top:** A section with undesired expression marked for removal. **Bottom:** The user chooses a desired frame and its expression coefficients are defined fixed for the mid frame (red). The expression coefficients are linearly interpolated in the two remaining gaps and filled using our method.

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 [99]. Lastly, our method does not perform well for large pose variations in which previously occluded part of the head would need to be synthesized.

We compare our method with a single image fitting method described in [112]. This method also decomposes 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.* [34] 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 Figure 4.10 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 Figure 4.10 bottom, our method is able to fit the face more accurately.

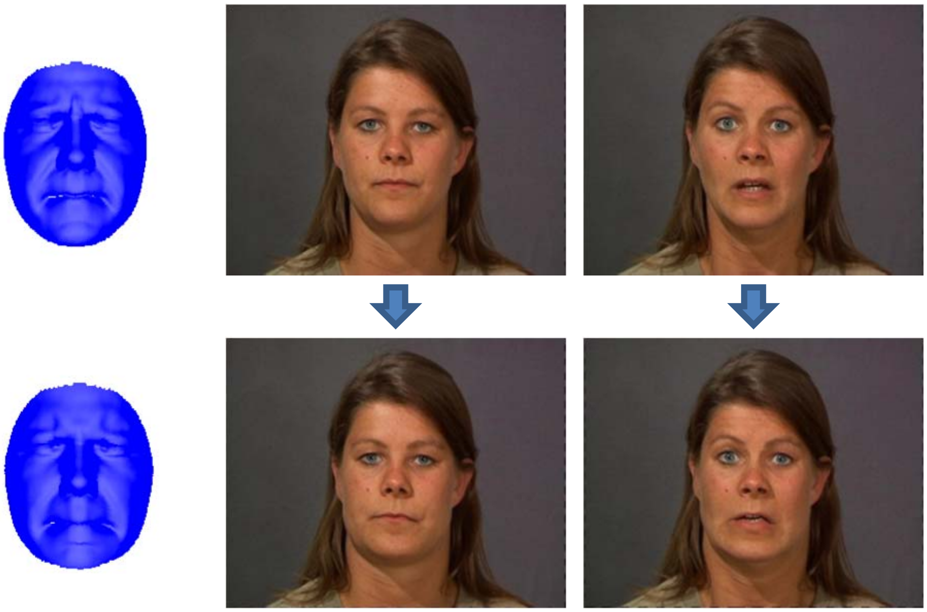

Figure 4.9: Changing identity coefficients γ results in changing the face structure throughout the video, independent of expression changes. **Top:** An original shape and frames. **Bottom:** After changing γ to a different person with wider chin.

4.5 Conclusion

In this chapter 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.

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.

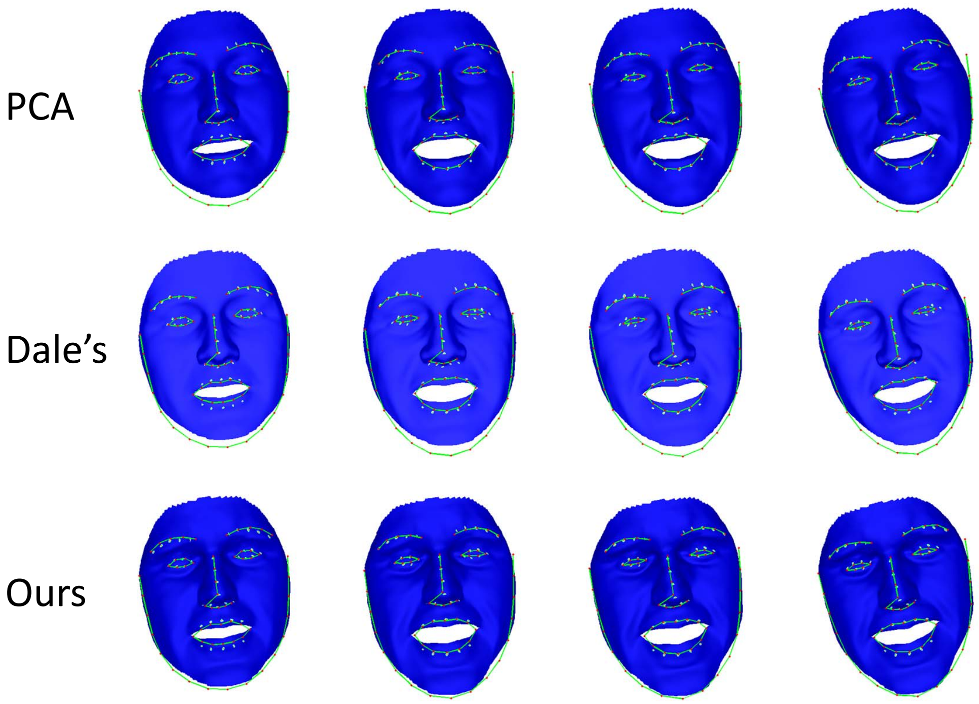

Figure 4.10: Comparison of the fitting result for frames 10,20,30,40. **Top:** Single image fitting of [112], which applies PCA model independently for each frame. **Middle:** An approximation of Dale *et al.* [34] using our method where we fit the identity coefficients using only the first frame. **Bottom:** Our method, which jointly fits all frames and enforces a common global identity. We are able to fit the face more accurately.

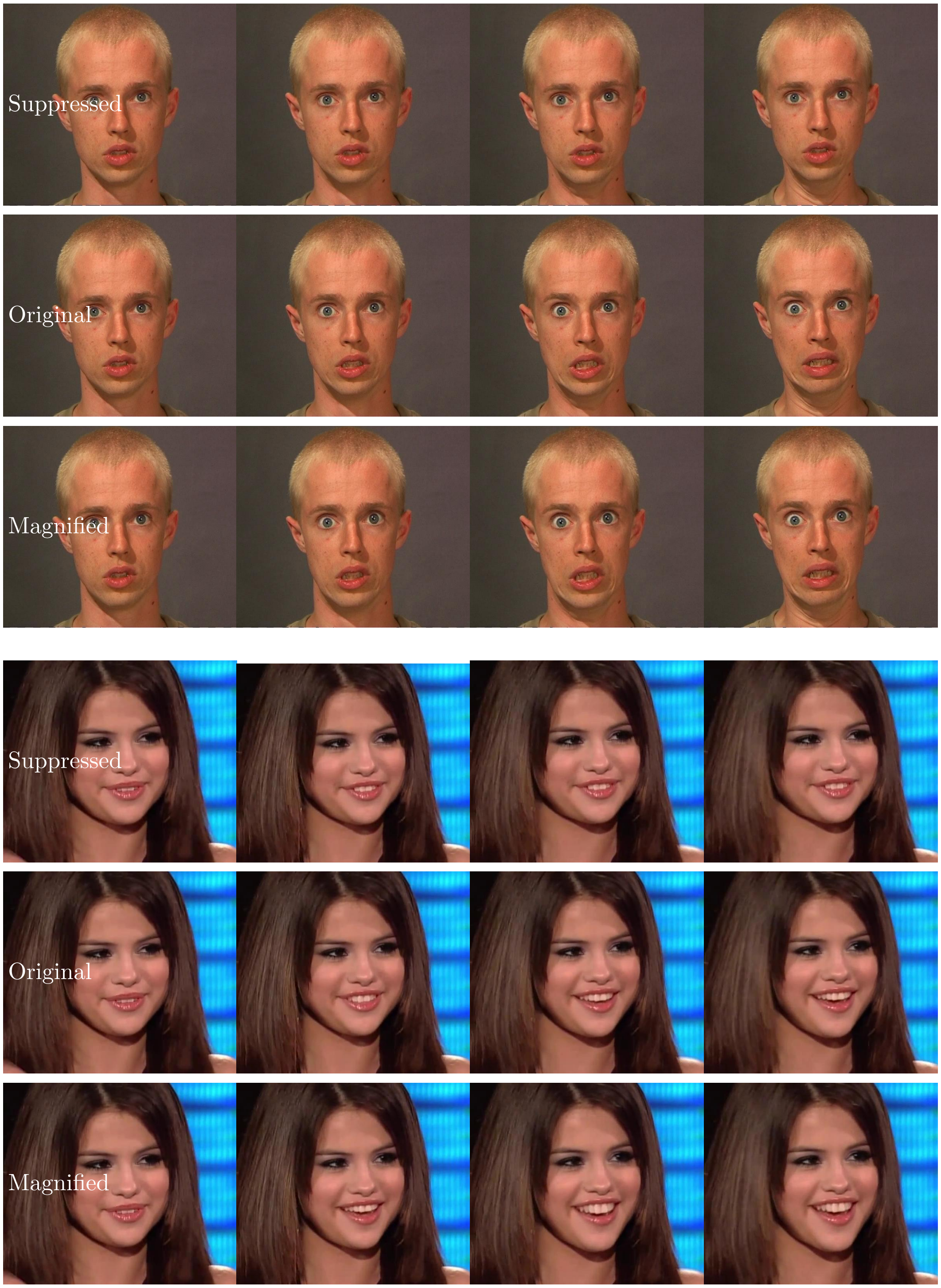

Figure 4.11: More examples of magnifying expressions, such as smile or fear. The original frames are in the middle rows. The synthesized expressions are shown above (suppressed) and below (magnified).

Chapter 5

Face Morphing

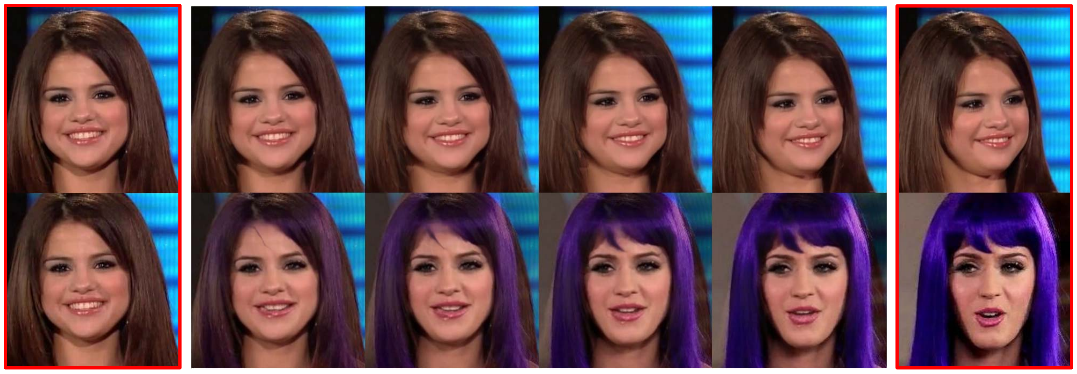

Figure 5.1: Our system can generate fully automatically high quality face morphing animation between faces of different pose and expression. **Top:** morphing between images of the same subject. **Bottom:** morphing between different subjects. The input images are the first and last column (highlighted in red).

Image morphing is a special visual effect in which one image is smoothly transformed into another. It has been extensively explored and widely used in motion pictures and animations. Image morphing between two images usually begins with extracting features from both images and building correspondence between the two feature sets. A pixel-wise mapping function is then derived from the sparse feature correspondence, which is used to warp both input images into desired alignment at each interpolation position. Finally, color interpolation is performed to generate each transition frame [107].

In this chapter we study the problem of image morphing between two face images, either from the same or different individuals. This is a challenging task, since human faces are highly non-rigid and could perform large 3D shape deformation under expression and pose variations. Moreover, human perception is sensitive to even a small amount of artifacts in faces. The challenge is significantly greater when the two input images have a large difference in both pose and expression.

Generic image morphing methods do not employ a 3D face model and, therefore, they are unable to accurately factor the differences between the two images due to pose and expression variations. As we will demonstrate later, without such accurate models of pose and expression, the interpolation results become unnatural. Furthermore, some previous morphing methods require tedious manual labeling on input images, which is undesirable in many applications.

We propose a novel approach for generating high quality face morphing animations. Our system is fully automatic and requires only the two end images as input. We first extract facial landmarks from each input image, and project them on a subspace learned from an external face shape dataset to recover the 3D face geometry. The geometry is factored into the pose and the expression of the input face. These are interpolated independently to create realistic intermediate shapes of the input face. For expression interpolation, we combine the expression flow [112] derived from the interpolated 3D models. Finally, our system employs an iterative appearance optimization framework where each intermediate frame is required to have similar geometry to the two corresponding interpolated models (up to a small residual optical flow to capture subtle geometric changes), as well as similar appearance to its neighbor frames. This optimization results in a sharp and temporally coherent interpolation sequence, as shown in examples in Figure 5.1.

To demonstrate the effectiveness of the proposed algorithm we compare it against a number of commonly used morphing approaches. Our experiments indicate that the proposed approach can generate higher quality face morphing results. As an additional application, we show how the system can be used to delete an undesired expression from a video sequence.

5.1 Related work

User-assisted morphing. Most previous image morphing approaches require the user to manually specify feature correspondences between input images [106]. Mesh morphing methods [107, 60] define the spatial transformation at mesh points or snake curves. The mesh is then deformed with the constraint to maintain topological equivalence. The field morphing method [10] uses corresponding lines in the source and target images to define the mapping function between the two images, which simplifies the user input. These methods have already been implemented in commercial systems. However they require tedious annotation and do not work well for large variations in pose, as the 2D transformations used in these methods do not preserve 3D facial shape.

The view morphing method [85] preserves the 3D shape during morphing without the explicit use of 3D models. It works by pre-warping the two images prior to computing a morph, and then post-warping the interpolated images. This method works well for rigid objects. However, it cannot accurately estimate the projection parameters for a deformable object like a face exhibiting change in expression. Furthermore, it still requires substantial manual correspondence. In contrast, our approach is fully automatic and can handle both pose and expression variations.

Automatic morphing. Bichsel [11] proposed a fully-automatic morphing technique using a Bayesian framework and maximizing the penalized likelihood of the spatial and color transformations given the input images. Zanella et al. [115] and some other commercial face morphing packages such as the Face Morpher[1] used Active Shape Model (ASM) [27] to find corresponding points and then linearly interpolate their locations for mostly frontal view morphing. Our method also uses ASM for initial correspondence, however this is followed by projection on a 3D subspace and further appearance optimization. Therefore our method is more robust to the inaccuracies in point locations, and can handle large 3D pose variations.

Mahajan et al. proposed the Moving Gradients [67] method for automatic image interpolation which handles occlusions explicitly. It finds an optimal path for gradients at every pixel from one image to the other, and get impressive results. However, the motions were roughly linear and thus mostly confined to small 2D changes. Another

[1] http://www.facemorpher.com/

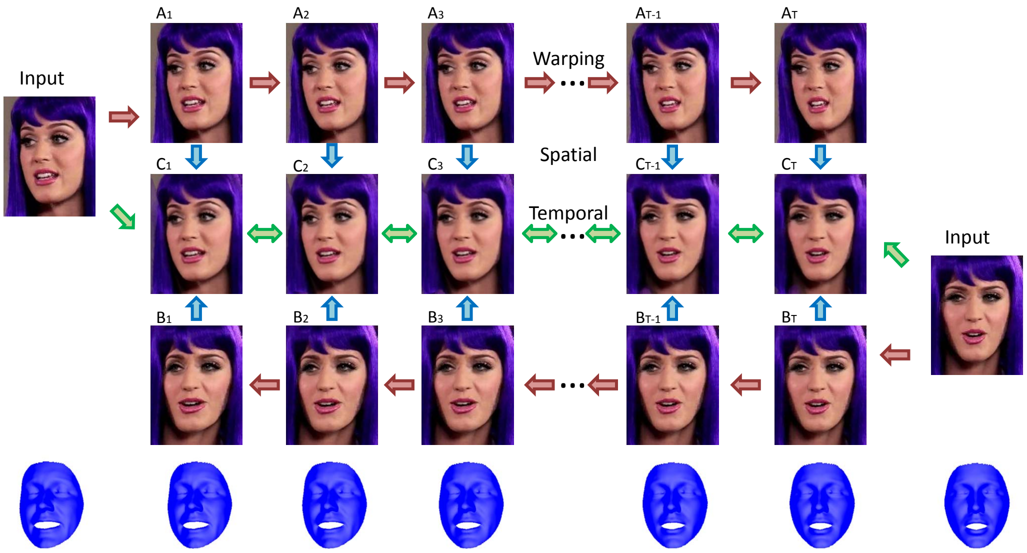

Figure 5.2: Overview of our framework. We first fit 3D shapes to both input images, and interpolate the 3D models for intermediate frames. The faces are warped using the difference between the 3D models. In each frame, the warped faces are blended together.

recently proposed automatic method is Regenerative Morphing (RM) [87] in which the output sequence is regenerated from small pieces of the two source images in a patch-based optimization framework. The method generates appealing automatic morphs between radically different images and can produce impressive image interpolation results with additional point correspondences (either manual or based on automatic feature matching). Our optimization algorithm was inspired by RM, but it operates at the frame-level, as opposed to the patch-level. We report comparisons to RM, using the same ASM correspondences as our method, as well as to other general morphing methods (Sec. 5.4). The results show that our method is better suited for high-quality realistic face morphing.

5.2 Algorithm overview

The framework of the proposed face morphing approach is shown in Figure 5.2. Given the two input images A and B, we first fit a 3D shape to each of them, as described in Section 5.2.1. A 3D shape contain two sets of parameters: external parameters describing the 3D pose of the face, and intrinsic parameters describing the facial geometry of the person under the effect of facial expression. We then linearly interpolate both the intrinsic and external parameters of the two input faces, resulting in a series of interpolated 3D face models, as shown in the bottom of Figure 5.2.

We use the dataset of 3D face models proposed by Vlasic et al. [99]. Since all the 3D models have one-to-one dense vertex correspondence, by projecting the 3D models to the image plane, we obtain warping functions from each input image to all interpolated frames. By warping the input image A with its corresponding warping functions, we can generate a series of deformed images $A_1, A_2, ..., A_T$, where T is the number of intermediate frames to be generated. Similarly, a series of images $B_1, B_2, ..., B_T$ can be obtained by warping image B.

A weighted averaging between A_t and B_t would give us an interpolated face C_t as the morphing result, as shown in Figure 5.2. However this approach is not optimal, as each A_t and B_t pairs is processed individually, thus they are not necessarily well aligned and temporal coherence is not guaranteed. To further improve the quality of the morphing sequence, inspired by the recent Regenerative Morphing approach [87], we define a morphing energy function and employ an iterative optimization approach to minimize it, as described in Section 5.3.

5.2.1 Fitting 3D shapes

We first describe how to fit a 3D face shape to a single face image. We follow the method described in the Expression Flow system [112], which is an efficient method for fitting 3D models to near-frontal face images. This method first localizes facial landmarks using the Active Shape Model (ASM) [71], then fits the 3D model based on these landmarks.

The face shape is defined using a shape vector s concatenating the X, Y, Z coordinates of all vertices. The deformable face model is constructed by running Principal Component Analysis (PCA) [17] on the training dataset from Vlasic et al. [99]. A new shape can be formed as a linear combination of eigenvectors v_i and the mean shape $\bar{s}$:

$$s = \bar{s} + \sum \beta_i v_i = \bar{s} + \mathbf{V}\beta. \tag{5.1}$$

The 3D fitting is performed by varying the coefficients β in order to minimize the error between the projections of the pre-defined landmarks on the 3D face geometry and the 2D feature points detected by ASM. The fitting error for the kth landmark is defined as:

$$E_k = \frac{1}{2}||P \cdot L_k \cdot (\bar{s} + \mathbf{V}\beta) - X_k||^2 \qquad (5.2)$$

where P is a 2×3 projection matrix, L_k is a selection matrix that selects the vertex corresponding to the kth landmark and X_k are the X, Y coordinates of the kth ASM landmark. The total energy E, which is the total fitting error of all landmarks, is minimized with respect to the projection matrix P and the shape coefficients β using iterative optimization approach. In the first step the projection matrix P is optimized to align the current 3D shape to the 2D features. In the second step the shape coefficients β are optimized to deform the 3D shape for better fitting. The two steps are repeated until convergence.

5.2.2 Finding projection matrix P

In the 3D fitting algorithm described above the projection matrix P is computed to align the current 3D shape to the 2D landmarks. The method in the Expression Flow system [112] uses a weak perspective projection model, and solves P using a least squares approach. However, we find this approach not optimal, since the variations of facial shapes are also captured by P, resulting in an inaccurate solution of β. For instance, an elongated face in the image can be explained either by a large scale in the Y direction of the projection matrix, or by an elongated 3D face geometry.

To avoid this ambiguity, we add two constraints to the projection matrix: the X and Y directions have the same scale, and there is no skew between them. Such a camera model is usually referred to as a restricted camera model [50]. The projection matrix P is parameterized with 6 variables, which can be estimated iteratively using the Levenberg-Marquardt algorithm [73].

5.3 Optimization of shape and appearance

We pose morphing as an optimization problem with an objective function that requires each frame in the output sequence of faces to be similar in shape (expression and pose) and appearance to a linear interpolation of these from the two sources. In addition, we want the shape and appearance to change smoothly from frame to frame.

Given the 3D shapes for the faces in input images A and B, we can easily interpolate their expression and 3D pose parameters to any intermediate view using an image warp induced from the 3D change. After that we employ an iterative optimization that maximizes the appearance similarity of the morphed sequence to the shape interpolated views while maintaining temporal coherence.

5.3.1 Prewarping the sources

The 3D face shapes contain two sets of parameters. The intrinsic parameters are the shape coefficients β, describing the facial geometry of a person exhibiting a facial expression. The external parameters include the rotation angles $\theta_x, \theta_y, \theta_z$, the scale c, and the 2D translations x_0, y_0, describing the 3D pose of the face. We linearly interpolate the intrinsic and external parameters of two input faces, resulting in a series of interpolated 3D face models. The interpolation of the each parameter p in frame t is defined as:

$$p_t = k_A p_A + k_B p_B, \quad (t = 1, ..., T) \tag{5.3}$$

where $k_A = (1 - \frac{t}{1+T})$ and $k_B = \frac{t}{1+T}$. Then we reconstruct the 3D shapes for all intermediate frames, as shown in Figure 5.2 (bottom row).

To warp the input faces to the desired pose and expression, we apply “Expression Flow” [112]. As shown in Figure 5.3, given two 3D shapes, we compute the difference between the projections of each corresponding vertex, and get a flow map. Applying the flow warps the original faces A and B to images A_t and B_t correspondingly with an interpolated pose and expression.

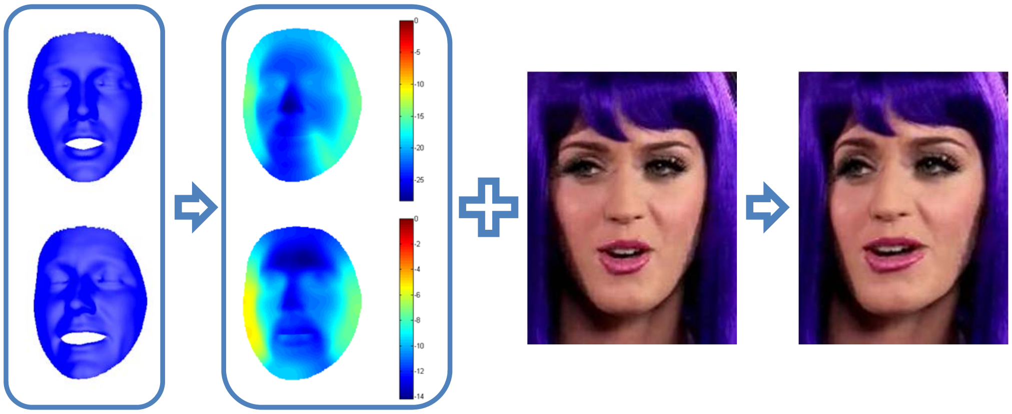

Figure 5.3: Warping the input image using expression flow. The flow map is computed by comparing the difference between two 3D models. The resulting image is the face warped to new pose and expression.

5.3.2 Appearance optimization

A simple weighted averaging between A_t and B_t would give us an interpolated face C_t, as shown in Figure 5.2. However this approach is not optimal, as each A_t and B_t pair is prewarped individually, thus temporal coherence is not guaranteed. Furthermore, C_t may be blurry due to misalignment between A_t and B_t. To further improve the quality of the morphing sequence, inspired by the Regenerative Morphing method [87], we define a morphing energy function and employ an iterative optimization to minimize it.

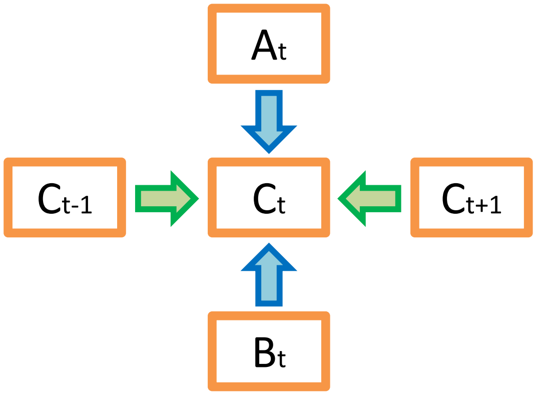

Figure 5.4: Appearance optimization.

First, every frame should be similar to the source images with an interpolated pose and expression. This prevents the appearance of the sequence from deviating too much from the source images. Second, the changes between every frame and its interpolated neighbor frames should be small. These two requirements give the following energy function in frame t.

$$\begin{aligned} E_t &= k_t^{(A)}||C_t - W_{f_{A_t}}(A_t)||^2 + k_t^{(B)}||C_t - W_{f_{B_t}}(B_t)||^2 \\ &+ k^{(C)}||C_t - W_{f_{C_{t-1}}}(C_{t-1})||^2 + k^{(C)}||C_t - W_{f_{C_{t+1}}}(C_{t+1})||^2 \end{aligned}$$

where A_t and B_t are input faces warped by expression flow and C_t is the interpolated face at frame t. In order to handle small residual misalignments between the warped sources, we apply a residual warp W induced by an optical flow [62] between A_t (or B_t, C_{t-1}, C_{t+1}) and C_t. We set $k^{(C)}$ to 1, and then $k_t^{(A)}$ and $k_t^{(B)}$ are interpolated between 0 and 1.

This quadratic energy function is minimized by a simple weighted sum of four images. Hence the face at frame t is updated as follows:

$$\begin{aligned} C_t &= k_t^{(A)}W_{f_{A_t}}(A_t) + k_t^{(B)}W_{f_{B_t}}(B_t) \\ &+ k^{(C)}W_{f_{C_{t-1}}}(C_{t-1}) + k^{(C)}W_{f_{C_{t+1}}}(C_{t+1}) \end{aligned} \tag{5.4}$$

We run typically two iterations for each frame and sweep over all frames in consecutive order for three times back and forth for a convergence of the entire morph sequence. Our algorithm is summarized in Algorithm 9.

5.3.3 Warping background

The optimization above generates a morph sequence for the face region only. In addition, we apply optical flow based interpolation to warp the background outside the face region, by interpolating the flow for each frame. We use the Moving Least Squares [84] method to smoothly blur the difference between the two flows at the boundary between the regions.

Algorithm 9 *Appearance Optimization*

```
Fit 3D shapes to two input images $A$ and $B$.
Compute interpolated 3D face shapes.
for all frames $t$ do
    Warp input images $A$ and $B$ to interpolated states $A_t$ and $B_t$.
end for
Initialize $C_t$ as weighted sum of $A_t$ and $B_t$.
repeat
    for all frames $t$ do
        repeat
            Optimize $C_t$ using Eq. 5.4.
        until frame converges
    end for
until sequence converges
```

5.4 Experiments

5.4.1 Face morphing

We first apply the proposed method in face morphing between images of the same subject. The results on two pairs of images are shown in Figure 5.8, 5.9 and 5.10 with closed-up views. We compare our method with four previous morphing methods: registration-based crossfading, mesh morphing, optical flow, and regenerative morphing. For cross-fading, we first register the two input images by estimating a similarity transform between them using the detected feature points. This is similar to the frame transition method used in the Photobios system [58]. For mesh morphing, we triangulate the face image using the 68 feature points detected by the ASM method. We use the recently proposed optical flow method [62] for comparison. For regenerative morphing, we use the authors' implementation.

As shown in the figures, the face morphing results generated by previous methods contain noticeable artifacts, mainly due to the large pose and expression changes between input images. The results generated by our system have fewer artifacts and are of higher quality. We also noticed that results generated by regenerative morphing

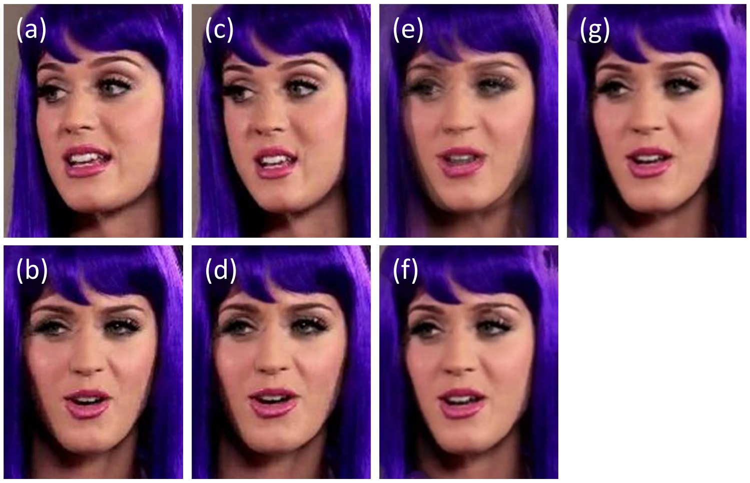

Figure 5.5: The result of disabling appearance optimization. (a) and (b) and two input images. (c) and (d) are acquired by warping the input images using 3D models. (e) is the result when using only similarity to the warped sources A_t and B_t (a weighted sum of (c) and (d)). (f) is the result when using only temporal smoothness (C_{t-1} and C_{t+1}). (g) is the result of the full system.

contain some temporal jittering, which is not visible in still images. Please refer to the supplementary video for temporal coherence comparison, as well as more results. We also apply face morphing between different subjects. One example is shown in Figure 5.1. Another example is shown in Figure 5.6, where we also compare the quality of interpolated faces with previous approaches.

To further evaluate the effectiveness of the proposed method, we compare our results against those generated by disabling appearance optimization. The 3D models are still used to warp the input images to the desired 3D pose and expression. As shown in Figure 5.5(e), we take the weighted sum of the warped faces. The results are blurry because the faces are not aligned well. Figure 5.5(f) shows the result when using only temporal smoothness (C_{t-1} and C_{t+1}), which has artifacts above the right eye. After applying appearance optimization, our system can get good intermediate frames.

5.4.2 Replacing undesired expression

Our method can be used to stitch the video after removing a portion of it. This can be used to remove an undesired expression, such as a tic or a yawn. As shown in Figure 5.7,

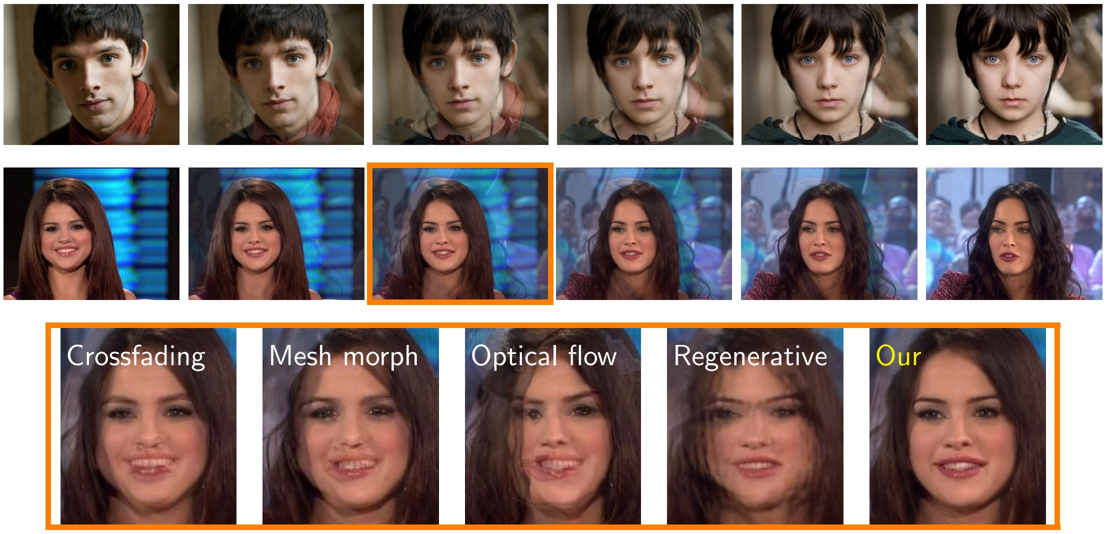

Figure 5.6: Morphing between different subjects. **Top two:** Our morphing result. **Bottom:** Comparison of different algorithms on the third frame.

the frames highlighted in red are manually identified and removed. Our face morphing system could generate a smooth transition and stitch the video after removing a clip.

5.4.3 Discussion

Our system is implemented in Matlab. It takes about 10 minutes to create 8 intermediate frames for face regions about 200 by 200 pixels, on a Intel CPU of 2.40 GHz. A large portion of the time is taken by the repeated optical flow computations. With latest GPU based optical flow method, the running time may be significantly reduced.

To be able to perform face morphing fully automatically, our system must rely on fully automatic performance of its components. Specifically, we rely on ASM for localizing facial components, and current ASM implementations can fail under large viewpoint variations or occlusions. We use optical flow to capture subtle geometric changes and build a residual warp. However, it might break when there are large differences in facial appearance, skin tones or illumination between two inputs. Therefore the current method is not effective for morphing between radically different people.

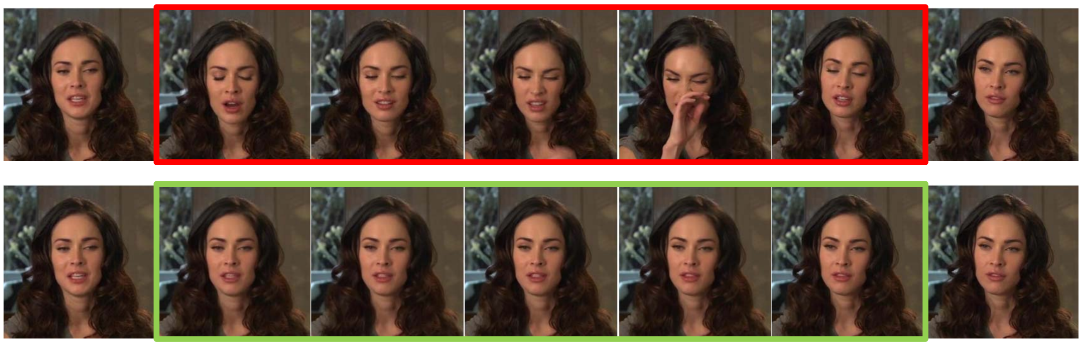

Figure 5.7: Our method can stitch the video after removing a clip. **Top:** the original video. The frames to be removed are highlighted in red. **Bottom:** the result. The new frames interpolated are highlighted in green. In this example we align the before/after frames for clarity, but the padding need not have the same duration as the removed clip.

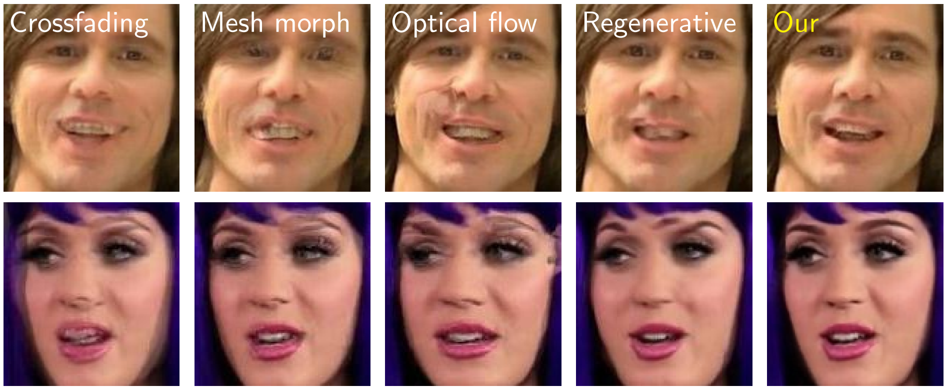

Figure 5.8: The close-up views of the middle column of Figure 5.9 and 5.10.

5.5 Conclusion

We address the problem of generating high quality face morphing animation given two faces of difference pose and expression. We show that a 3D subspace model learned from a small collection of human faces exhibiting realistic expression, constrains well the space of possible face deformations for interpolating casual face photos. Unlike traditional warping methods used for morphing that require accurate correspondence between the two source faces, we warp the two faces independently and only roughly

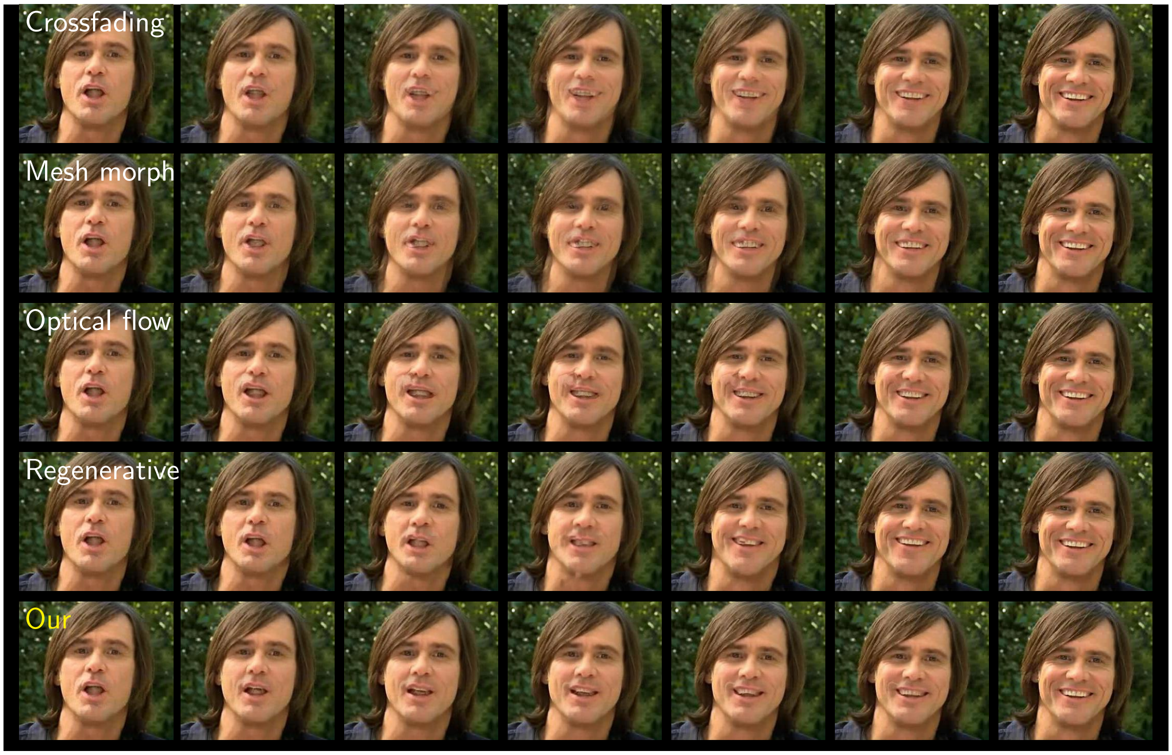

Figure 5.9: Face morphing results of a male subject. **Top row**: crossfading. **2nd row**: mesh morphing. **3rd row**: simple optical flow. **4th row**: regenerative morphing. **Bottom row**: our method. The close-up views of the middle columns are shown in Figure 5.8.

using an 3D model based flow. Thus, we avoid traditional warping artifacts like fold-overs and “holes”. The appearance optimization can recover small misalignment and other small changes not covered by the shape model and so traditional blurriness and hosting artifacts are suppressed. Future direction include handling partial occlusion (like sunglasses) using a more robust appearance optimization, face extrapolation, non-linear interpolation and applying this approach to other classes of objects for which effective 3D subspace models can be learned.

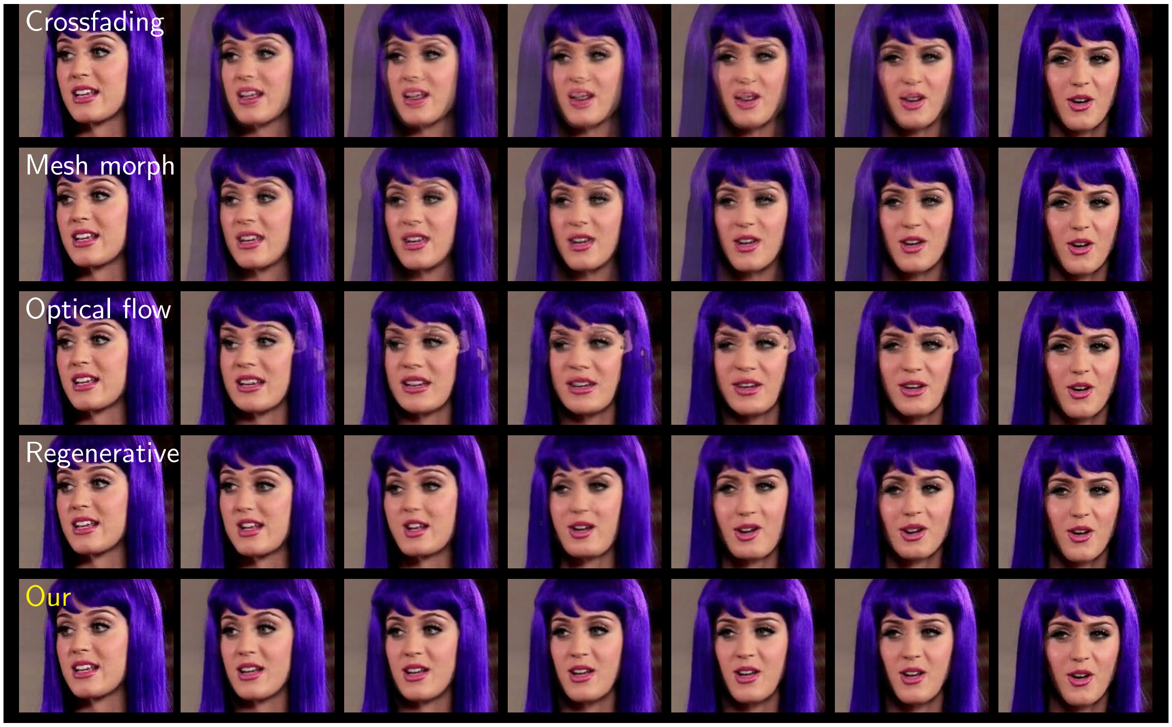

Figure 5.10: Face morphing results of a female subject. **Top row**: crossfading. **2nd row**: mesh morphing. **3rd row**: simple optical flow. **4th row**: regenerative morphing. **Bottom row**: our method. The close-up views of the middle columns are shown in Figure 5.8.

Chapter 6

Conclusion and Future Work

In this dissertation, we aim to develop a framework that could automatically edit facial expression in photos and videos in a semantic level. The proposed method is non-intrusive, computationally efficient, and can be used with existing photos or videos. Our system consists of four major components: facial feature localization, face model fitting, expression flow computation, and image compositing.

We introduce three different applications: facial component transfer, expression editing, and face morphing. For facial component transfer, we address the problem of correcting an undesirable expression on a face photo by transferring local facial components from another face photo of the same person which has the desired expression. We present Expression Flow, a 2D flow field which is computed by projecting the difference between the two 3D shapes back to 2D. It describes how to warp the target face photo to match the expression of the reference photo. User studies suggest that our system is able to generate face composites with much higher fidelity than existing methods.

For expression editing, we address the problem of editing facial expression in video, such as exaggerating, attenuating or replacing the expression with a different one. 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. 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.

For face morphing, we propose a new face morphing approach that explicitly deals with large pose and expression variations. We recover the 3D face geometry of the input images using a projection on a pre-learned 3D face subspace. The geometry is interpolated by factoring the expression and pose and varying them smoothly across the sequence. We pose the morphing problem as an iterative optimization with an objective that combines similarity of each frame to the geometry-induced warped sources, with a similarity between neighboring frames for temporal coherence. Experimental results show that our method can generate higher quality face morphing results for more extreme pose, expression and appearance changes than previous methods.

Our work can be extended in a few ways. First, Our system currently uses the face expression dataset collected by Vlasic et al. Although our system can work reliably well on a wide variety of people, we are also aware that the dataset is not rich enough to handle all possible expressions. As future work we plan to use existing 3D face capturing methods to capture more data to enrich our dataset. Second, faces of various expressions lie in a nonlinear manifold, which may not be well represented by multilinear subspaces. A future direction is to explore joint nonlinear manifolds to model shape variations.

References

[1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. In *ACM SIGGRAPH*, volume 23, pages 294–302, 2004.

[2] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Processing*, 54(11):4311–4322, 2006.

[3] S. Asteriadis, N. Nikolaidis, A. Hajdu, and I. Pitas. An eye detection algorithm using pixel to edge information. In *International Symposium on Control, Communications, and Signal Processing*, 2006.

[4] A. Azarbayejani, B. Horowitz, and A. Pentland. Recursive estimation of structure and motion using relative orientation constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 294–299, 1993.

[5] A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland. Visually controlled graphics. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(6):602–605, 1993.

[6] L. Bai, L. Shen, and Y. Wang. A novel eye location algorithm based on radial symmetry transform. In *International Conference on Pattern Recognition*, pages 511–514, 2006.

[7] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[8] T. Baltrusaitis, P. Robinson, and L. Morency. 3d constrained local model for rigid and non-rigid facial tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2610–2617, 2012.

[9] C. Basso, T. Vetter, and V. Blanz. Regularized 3d morphable models. In *IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis*, pages 3–10, 2003.

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

[11] M. Bichsel. Automatic interpolation and recognition of face images by morphing. In *International Conference on Automatic Face and Gesture Recognition*, 1996.

[12] D. Bitouk, N. Kumar, S. Dhillon, P. Belhumeur, and S. K. Nayar. Face swapping: automatically replacing faces in photographs. In *ACM SIGGRAPH*, pages 1–8, 2008.

[13] M. J. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *International Conference on Computer Vision*, pages 374–381, 1995.

[14] M. J. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision*, 25(1):23–48, 1997.

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

[16] V. Blanz, K. Scherbaum, T. Vetter, and H. P. Seidel. Exchanging faces in images. *Computer Graphics Forum*, 23(3):669–676, 2004.

[17] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. *ACM Trans. Graphics (SIGGRAPH)*, pages 187–194, 1999.

[18] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23, 2001.

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

[20] Q. Cai, D. Gallup, C. Zhang, and Z. Zhang. 3D deformable face tracking with a commodity depth camera. In *European Conference on Computer Vision*, pages 229–242, 2010.

[21] P. Campadelli, R. Lanzarotti, and G. Lipori. Precise eye localization through a general-to-specific model definition. In *British Machine Vision Conference*, page 187, 2006.

[22] P. Campadelli, R. Lanzarotti, and G. Lipori. Precise eye and mouth localization. *International Journal of Pattern Recognition and Artificial Intelligencece*, 23(3):359–377, 2009.

[23] E. J. Candès, J. K. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Information Theory*, 52(2):489–509, 2006.

[24] L. Chen, L. Zhang, L. Zhu, M. Li, and H. Zhang. A novel facial feature localization method using probabilistic-like output. In *Asian Conference on Computer Vision*, 2004.

[25] D. Comaniciu, P. Meer, and S. Member. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24:603–619, 2002.

[26] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[27] 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.

[28] T. F. Cootes. `http://personalpages.manchester.ac.uk/staff/timothy.f.cootes/data/tarfd_markup/tarfd_markup.html`.

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

[30] 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.

[31] T. F. Cootes, G. Wheeler, K. Walker, and C. Taylor. View-based active appearance models. *Image and Vision Computing*, 20(9):657–664, 2002.

[32] D. Cristinacce, T. Cootes, and I. Scott. A multi-stage approach to facial feature detection. In *British Machine Vision Conference*, 2004.

[33] D. Cristinacce and T. F. Cootes. Feature detection and tracking with constrained local models. In *British Machine Vision Conference*, pages 929–938, 2006.

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

[35] D. DeCarlo and D. Metaxas. The integration of optical flow and deformable models with applications to human face shape and motion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 231–238, 1996.

[36] D. Decarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *International Journal of Computer Vision*, 38(2):99–127, 2000.

[37] R. Dovgard and R. Basri. Statistical symmetric shape from shading for 3d structure recovery of faces. In *European Conference on Computer Vision*, pages 99–113, 2004.

[38] M. Elad and M. Aharon. Image denoising via learned dictionaries and sparse representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 895–900, 2006.

[39] M. Everingham and A. Zisserman. Regression and classification approaches to eye localization in face images. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 441–448, 2006.

[40] G. Faigin. *The Artist's Complete Guide to Facial Expression.* Watson-Guptill Publications Inc., New York, 1991.

[41] G. Fanelli, T. Weise, J. Gall, and L. J. V. Gool. Real time head pose estimation from consumer depth cameras. In *DAGM-Symposium*, pages 101–110, 2011.

[42] Z. Farbman, G. Hoffer, Y. Lipman, D. Cohen-Or, and D. Lischinski. Coordinates for instant image cloning. In *ACM SIGGRAPH*, pages 1–9, 2009.

[43] R. Fattal. Edge-avoiding wavelets and their applications. In *ACM SIGGRAPH*, volume 28, 2009.

[44] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61, 2003.

[45] X. Gao, Y. Su, X. Li, and D. Tao. A review of active appearance models. *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40(2):145–158, 2010.

[46] C. Garcia and M. Delakis. Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Trans. Pattern Analyssi and Machine Intelligence*, 26(11):1408–1423, 2004.

[47] L. Gu and T. Kanade. A generative shape regularization model for robust face alignment. In *European Conference on Computer Vision*, pages 413–426, 2008.

[48] M. Hamouz, J. Kittler, J. Kamarainen, P. Paalanen, and H. Kälviäinen. Affine-invariant face detection and localization using gmm-based feature detector and

enhanced appearance model. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 67–72, 2004.

[49] M. Hamouz, J. Kittler, J. Kamarainen, P. Paalanen, H. Kälviäinen, and J. Matas. Feature-based affine-invariant localization of faces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(9):1490–1495, 2005.

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

[51] P. S. Huang, C. Zhang, and F.-P. Chiang. High-speed 3-D shape measurement based on digital fringe projection. *Optical Engineering*, 42(1):163–168, 2003.

[52] Y. Huang, Q. Liu, and D. Metaxas. A component based deformable model for generalized face alignment. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[53] O. Jesorsky, K. J. Kirchberg, and R. Frischholz. Robust face detection using the hausdorff distance. In *International Conference Audio- and Video-Based Biometric Person Authentication*, pages 90–95, 2001.

[54] N. Joshi, W. Matusik, E. H. Adelson, and D. J. Kriegman. Personal photo enhancement using example images. *ACM Trans. Graphics*, 29:12:1–12:15, 2010.

[55] A. Kanaujia and D. Metaxas. Recognizing facial expressions by tracking feature shapes. In *International Conference on Pattern Recognition*, pages 33–38, 2006.

[56] I. Kemelmacher-Shlizerman and R. Basri. 3d face reconstruction from a single image using a single reference face shape. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33:394–405, 2011.

[57] I. Kemelmacher-Shlizerman, A. Sankar, E. Shechtman, and S. M. Seitz. Being John Malkovich. In *European Conference on Computer Vision*, pages 341–353, 2010.

[58] I. Kemelmacher-Shlizerman, E. Shechtman, R. Garg, and S. M. Seitz. Exploring photobios. In *ACM SIGGRAPH*, page 61, 2011.

[59] R. Kothari and J. Mitchell. Detection of eye locations in unconstrained visual images. In *International Conference on Image Processing*, pages 519–522, 1996.

[60] S. Lee, G. Woberg, K.-Y. Chwa, and S. Y. Shin. Image metamorphosis with scattered feature constraints. *IEEE Trans. Visualization and Computer Graphics*, 2(4):337–354, 1996.

[61] T. Leyvand, D. Cohen-Or, G. Dror, and D. Lischinski. Data-driven enhancement of facial attractiveness. In *ACM SIGGRAPH*, pages 1–9, 2008.

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

[63] C. Liu, H.-Y. Shum, and W. T. Freeman. Face hallucination: Theory and practice. *International Journal of Computer Vision*, 75:115–134, 2007.

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

[65] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *DARPA Image Understanding Workshop*, pages 121–130, 1981.

[66] Y. Ma, X. Ding, Z. Wang, and N. Wang. Robust precise eye location under probabilistic framework. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 339–344, 2004.

[67] D. Mahajan, F.-C. Huang, W. Matusik, R. Ramamoorthi, and P. Belhumeur. Moving gradients: a path-based method for plausible image interpolation. *ACM Trans. Graphics*, 28:1–11, 2009.

[68] A. Martinez and R. Benavente. The AR face database. *CVC Technical Report 24*, June 1998.

[69] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.

[70] I. Matthews, J. Xiao, and S. Baker. 2D vs. 3D deformable face models: Representational power, construction, and real-time fitting. *International Journal of Computer Vision*, 75(1):93–113, 2007.

[71] S. Milborrow and F. Nicolls. Locating facial features with an extended active shape model. In *European Conference on Computer Vision*, pages 504–513, 2008.

[72] Z. Niu, S. Shan, S. Yan, X. Chen, and W. Gao. 2d cascaded adaboost for eye localization. In *International Conference on Pattern Recognition*, pages 1216–1219, 2006.

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

[74] A. P. Pentland, B. Moghaddam, and T. E. Starner. Viewbased and modular eigenspaces for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[75] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM SIGGRAPH*, pages 313–318, 2003.

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

[77] F. Pighin, R. Szeliski, and D. H. Salesin. Resynthesizing facial animation through 3d model-based tracking. In *International Conference on Computer Vision*, volume 1, pages 143–150, 1999.

[78] M. Proesmans, L. Van Gool, and A. Oosterlinck. One-shot active 3D shape acquisition. In *International Conference on Pattern Recognition*, volume 3, pages 336–340, 1996.

[79] S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3D morphable model. In *International Conference on Computer Vision*, pages 59–66, 2003.

[80] 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.

[81] 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.

[82] J. M. Saragih, S. Lucey, and J. F. Cohn. Face alignment through subspace constrained mean-shifts. In *International Conference on Computer Vision*, pages 1034–1041, 2009.

[83] J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, 91(2):200–215, 2011.

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

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

[86] S. Shan, Y. Chang, W. Gao, B. Cao, and P. Yang. Curse of misalignment in face recognition: Problem and a novel misalignment learning solution. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 314–320, 2004.

[87] E. Shechtman, A. Rav-Acha, M. Irani, and S. Seitz. Regenerative morphing. In *Conference on Computer Vision and Pattern Recognition*, San-Francisco, CA, 2010.

[88] Singular Inversions Inc. Facegen modeller manual. In *www.facegen.com*, 2009.

[89] J. Sung, T. Kanade, and D. Kim. Pose robust face tracking by combining active appearance models and cylinder head models. *International Journal of Computer Vision*, 80(2):260–274, 2008.

[90] K. Sunkavalli, M. K. Johnson, W. Matusik, and H. Pfister. Multi-scale image harmonization. In *ACM SIGGRAPH*, volume 29, 2010.

[91] X. Tan, F. Song, Z. Zhou, and S. Chen. Enhanced pictorial structures for precise eye localization under incontrolled conditions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1621–1628, 2009.

[92] X. Tan, F. Song, Z. Zhou, and S. Chen. Enhanced pictorial structures for precise eye localization under uncontrolled conditions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1621–1628, 2009.

[93] D. Terzopoulos and K. Waters. Analysis and synthesis of facial image sequences using physical and anatomical models. *Pattern Analysis and Machine Intelligence, IEEE Trans.*, 15(6):569–579, 1993.

[94] 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.

[95] J. A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Trans. Information Theory*, 50(10):2231–2242, 2004.

[96] M. Türkan, M. Pardàs, and A. E. Çetin. Human eye localization using edge projections. In *International Conference on Computer Vision Theory and Applications*, pages 410–415, 2007.

[97] R. Valenti and T. Gevers. Accurate eye center location and tracking using isophote curvature. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[98] Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *European Conference on Computer Vision*, 2002.

[99] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. In *ACM SIGGRAPH*, volume 24, pages 426–433, 2005.

[100] C. Vogler, Z. Li, A. Kanaujia, S. Goldenstein, and D. Metaxas. The best of both worlds: Combining 3D deformable models with active shape models. In *International Conference on Computer Vision*, pages 1–7, 2007.

[101] Y. Wang, X. Huang, C.-S. Lee, S. Zhang, Z. Li, D. Samaras, D. Metaxas, A. Elgammal, and P. Huang. High resolution acquisition, learning and transfer of dynamic 3-d facial expressions. In *EuroGraphics*, pages 677–686, 2004.

[102] Y. Wang, S. Lucey, and J. F. Cohn. Enforcing convexity for improved alignment with constrained local models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[103] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. *ACM Trans. Graphics*, 30(4):77, 2011.

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

[105] L. Williams. Performance-driven facial animation. In *ACM SIGGRAPH*, volume 24, pages 235–242, 1990.

[106] G. Wolberg. *Digital Image Warping.* IEEE Computer Society Press, Los Alamitos, CA, 1990.

[107] G. Wolberg. Recent advances in image morphing. In *Computer Graphics International*, 1996.

[108] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Analysis and Machine Itelligence*, 31(2):210–227, 2009.

[109] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2D+3D active appearance models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 535–542, 2004.

[110] F. Yang, J. Huang, and D. Metaxas. Sparse shape registration for occluded facial feature localization. In *IEEE International Conference on Automatic Face and Gesture Recognition and Workshops*, pages 272–277, 2011.

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

[112] F. Yang, J. Wang, E. Shechtman, L. Bourdev, and D. Metaxas. Expression flow for 3D-aware face component transfer. *ACM Trans. Graphics (SIGGRAPH)*, 27(3):60, 2011.

[113] A. Yuille and P. Hallinan. Active vision. chapter Deformable templates, pages 21–38. MIT Press, 1993.

[114] A. L. Yuille, P. W. Hallinan, and D. S. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, 1992.

[115] V. Zanella, H. Vargas, and L. V. Rosas. Active shape models and evolution strategies to automatic face morphing. In *International Conference on Adaptive and Natural Computing Algorithms*, pages 564–571, 2007.

[116] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime faces: High-resolution capture for modeling and animation. In *ACM Annual Conference on Computer Graphics*, pages 548–558, August 2004.

[117] S. Zhang, Y. Zhan, M. Dewan, J. Huang, D. Metaxas, and X. Zhou. Towards robust and effective shape modeling: Sparse shape composition. *Medical Image Analysis*, pages 265–277, 2012.

[118] Z. Zhang. Microsoft kinect sensor and its effect. *IEEE Multimedia*, 19(2):4–10, 2012.

[119] M. Zhou, L. Liang, J. Sun, and Y. Wang. AAM based face tracking with temporal matching and face segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 701–708, 2010.

[120] X. S. Zhou, D. Comaniciu, and A. Gupta. An information fusion framework for robust shape tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(1):115–129, 2005.

[121] Y. Zhou, L. Gu, and H. Zhang. Bayesian tangent shape model: Estimating shape and pose parameters via bayesian inference. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 109–116, 2003.

[122] Z.-H. Zhou and X. Geng. Projection functions for eye detection. *Pattern Recognition*, 37(5):1049–1056, 2004.

FEI YANG

110 Frelinghuysen Road, Piscataway, NJ 08854
yfalan@gmail.com ⋄ http://yangfei.net

EDUCATION

Department of Computer Science, Rutgers University 2007/09 - Present
Ph.D. in Computer Science (expected in 2013)

Institute of Computing Technology, Chinese Academy of Sciences 2003/09 - 2006/07
M.E. in Computer Science

Department of Computer Science, Tsinghua University 1999/09 - 2003/07
B.E. in Computer Science

INDUSTRIAL EXPERIENCE

Bell Labs Murray Hill, NJ
Researcher (internship) 2012/06 - 2012/12

- Developed algorithms to reconstruct images from a single-pixel-camera.
- Developed algorithms to detect moving objects from compressive measurements.

Adobe Systems Seattle, WA
Summer Internships 2010/05 - 2010/08, 2011/06 - 2011/09

- Developed algorithms to copy facial components to a different face image.
- Developed algorithms to exaggerate facial expressions, and to transfer expressions between subjects.
- Developed algorithms to construct 3D face geometry from 2D images.

Microsoft Beijing, China
Software Engineer 2006/07 - 2007/08

- Member of Windows Live Parental Control team.
- Member of team developed Web Content Filtering 2.0.
- Developed an automatic testing framework.

ACADEMIC EXPERIENCE

Rutgers University New Brunswick, NJ
Research Assistant 2009/05 - 2012/06

- Developed CBIM FaceTracker 2.0, which is a fully automatic system, that is able to track facial features, detect eye blinks, track lip motion, and recognize facial expressions in real time. The system has been demonstrated in many exhibitions, and was reported by BBC and KQED-TV.
- Developed a realtime fatigue detection system, reported by Philadelphia Inquirer.
- Developed algorithms to recognize facial signs in American Sign Language.
- Developed parallel algorithms for MRI image reconstruction.

Teaching Assistant 2007/09 - 2009/05

- CS110: Introduction to Computer Science. CS113: Software Methodology. CS534: Computer Vision.

Chinese Academy of Sciences Beijing, China
Research Assistant 2004/07 - 2006/07

- Member of team to develop JDL Face Recognition system. Won several competitions (C++).
- Developed an automatic system to measure head poses using electromagnetic sensors (C++).

- Developed face recognition algorithms by using LDA and SVM (C++, Matlab).

Institute of HCI, Tsinghua University Beijing, China
Summer Internship 2002/06 - 2002/09

- Developed automatic face detection methods (C++, Matlab).

PATENTS

1. L. Bourdev, E. Shechtman, J. Wang and F. Yang, "Methods and Apparatus for Face Fitting and Editing Applications", Adobe Systems, US Patent 20,130,121,409, 2013.
2. J. Wang, E. Shechtman, L. Bourdev and F. Yang, "Methods and Apparatus for Facial Feature Replacement", Adobe Systems, US Patent 20,130,129,141, 2013.

SELECTED PUBLICATIONS

1. F. Yang, H. Jiang, Z. Shen, W. Deng and D. Metaxas, "Adaptive Low Rank and Sparse Decomposition of Video Using Compressive Sensing", IEEE International Conference on Image Processing (ICIP), 2013.
2. X. Yu, F. Yang, J. Huang and D.N. Metaxas, "Explicit Occlusion Detection based Deformable Fitting for Facial Landmark Localization", Automatic Face and Gesture Recognition (FG), 2013.
3. J. Liu, B. Liu, S. Zhang, F. Yang, P. Yang, D.N. Metaxas and C. Neidle, "Recognizing Eyebrow Movements Using CRFs for Non-Manual Grammatical Marker Detection in ASL", Automatic Face and Gesture Recognition (FG), 2013.
4. X. Yu, S. Zhang, Z. Yan, F. Yang, J. Huang, N. Dunbar, M. Jensen, J.K. Burgoon and D.N. Metaxas, "Is Interactional Dissynchrony a Clue to Deception: Insights from Automated Analysis of Nonverbal Visual Cues", Hawaii International Conference on System Sciences (HICSS), 2013.
5. F. Yang, L. Bourdev, E. Shechtman, J. Wang and D.N. Metaxas, "Facial Expression Editing in Video", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012.
6. F. Yang, E. Shechtman, J. Wang, L. Bourdev and D.N. Metaxas, "3D-Aware Appearance Optimization for Face Morphing", Graphics Interface (GI), 2012.
7. F. Yang, X. Yu, J. Huang, P. Yang and D.N. Metaxas, "Robust Eyelid Tracking for Fatigue Detection", IEEE International Conference on Image Processing (ICIP), 2012.
8. F. Yang, J. Huang, X. Yu, X. Cui and D.N. Metaxas, "Robust Face Tracking With a Consumer Depth Camera", IEEE International Conference on Image Processing (ICIP), 2012.
9. J. Huang and F. Yang, "Compressed Magnetic Resonance Imaging Based on Wavelet Sparsity and Nonlocal Total Variation", IEEE International Symposium on Biomedical Imaging (ISBI), 2012.
10. X. Cui, Q. Liu, S. Zhang, F. Yang and D.N. Metaxas, "Temporal Spectral Residual for Fast Salient Motion Detection", NeuroComputing, 2012.
11. D.N. Metaxas, B. Liu, F. Yang, P. Yang, N. Michael and C. Neidle, "Recognition of Nonmanual Markers in American Sign Language Using Non-Parametric Adaptive 2D-3D Face Tracking", the 8th International Conference on Language Resources and Evaluation (LREC), 2012.
12. F. Yang, J. Wang, E. Shechtman, L. Bourdev and D.N. Metaxas, "3D-Aware Expression Flow for 2D Face Compositing", ACM Transactions on Graphics, (Proc. SIGGRAPH), 2011.
13. F. Yang, J. Huang, P. Yang and D.N. Metaxas,"Eye Localization through Multi-scale Sparse Dictionaries", IEEE Conference on Automatic Face and Gesture Recognition (FG), 2011.
14. F. Yang, J. Huang and D.N. Metaxas, "Sparse Shape Registration for Occluded Facial Feature Localization", IEEE Conference on Automatic Face and Gesture Recognition (FG), 2011.