

Morphable Face Reconstruction with Multiple Images

Ming Zhao Tat-Seng Chua Terence Sim
Department of Computer Science
National University of Singapore
{zhaom, chuats, tsim}@comp.nus.edu.sg

Abstract

Efficient 3D face reconstruction is very important for face animation and recognition. The slow speed of the 3D morphable model is due to the texture mapping. To improve the speed, we only use the shape matching to recover the 3D shape and use texture mapping to get the texture. However, only with the shape information, one image is not enough for accurate 3D face reconstruction. So we propose to use multiple images with the morphable shape model. First, with the feature points given on the multiple images, the 3D coordinates of the feature points are estimate by the pose estimation. Then, frontal and profile 2D morphable shape models are built to estimate the 3D morphable shape model. These two steps works iteratively to improve the result. At last, the texture is extracted from multiple images with the pose estimation from the estimated 3D face. The effectiveness of our method is demonstrated by the experimental results.

1. Introduction

3D face reconstruction from 2D images are very important for face animation, facial analysis and face recognition. However, it has been a challenging issue in computer graphics and computer vision literatures in the past decades. Since the pioneering work of Parke [13], many algorithms have been proposed for modelling the geometry of faces [5, 7, 4]. The 2D-based methods do not consider the specific structure of human faces, thus result in the poor performance on profile face images. In the work of Lam *et al.* [10], face samples with out-of-plane rotation are warped into frontal faces based on a cylinder face model, but it requires heavy manual labeling work. Shape from shading [18] has been explored to extract 3D face geometry information and generate virtual samples by rotating the generated 3D face models. However it requires that the face images are precisely aligned pixel-wise, which is difficult to be implemented in practice and even impossible for practi-

cal applications. The artificial 3D shape model proposed by Zhang *et al.* [12] developed a system to construct textured 3D face model from video sequence. Zhang's work needs two images close to the frontal view and two conditioned sequences including about 40 images, which are impractical for real applications.

The morphable 3D face model proposed by Blanz and Vetter *et al.* [1, 15] presented a 3D reconstruction algorithm to recover the shape and texture parameters based on a face image in arbitrary view. However, its speed can not satisfy the requirements of practical face recognition systems. Pighin *et al.* [14] used a generic face model and multiple images to recover the 3D face model. It can estimate the depth information by multiple images. However, with the generic face model, it need to specify many point to get accurate 3D model and can not correct the mis-registration errors. Recently, Hu and Jiang *et al.* [6, 8] presented an automatic 2D-to-3D integrated face reconstruction method to recover the 3D face model based on one frontal face image and it is much faster. However, Hu and Jiang's work can not accurately recover the depth information due to lacking the depth information.

The most time consuming part Vetter's work is the texture matching. It need to generate 3D texture in different pose and illumination. To improve the speed, we only tried to use the shape information. However, only with the shape information, one image is not enough for accurate 3D face reconstruction [6, 8]. So we propose to use multiple images with the morphable shape model. First, with the feature points labelled on the multiple images, the 3D coordinates of the feature points are estimate. Then, frontal and profile 2D morphable models are proposed to estimate the 3D morphable model. These two steps works iteratively to improve the result. At last, the texture is extracted from multiple images with the pose estimation from the estimated 3D face. Compared with Vetter's work, our method is more efficient; compared with Pighin's work, our method is more accurate and robust, and need fewer feature points; compared with Hu and Jiang's work, our method can recover the depth information more accurately, can handle arbitrary input im-

ages and have more realistic texture especially for the side view.

The rest of the paper is arranged as follows. The morphable shape modelling is described in Section 2. Section 3 discusses the pose estimation with multiple images, and the texture extraction with multiple images are given in Section 4. Section 5 presents the 3D reconstruction algorithm by morphable models with multiple images. Experimental results are provided in Section 6 before conclusions are drawn in Section 7.

2. Morphable Shape Modelling

Different from Vetter's 3D morphable model which includes both shape and texture information, our morphable model only utilize the shape information. This is key to the fast reconstruction. According to the 3D morphable shape model, different 2D morphable shape models can be induced, which extends the method by Hu and Jiang [6, 8] in that we can induce different kinds of 2D models instead of only one. As one 2D model can only accurately recover two coordinates, our extension can recover all the three coordinates. The 2D morphable models are used to recover the shape parameters of the 3D morphable shape model.

2.1. 3D Morphable Shape Model

We use the USF Human ID 3D database which includes 100 laser scanned heads [1] to build a 3D morphable face model. Each face model in the database has approximately 70,000 vertices. The number of the vertices is reduced to about 10623 for better performance in this paper.

The geometry of a 3D face model is represented with a shape vector $\mathbf{S} = (X_1, Y_1, Z_1, \dots, X_n, Y_n, Z_n)T \in \mathbf{R}^{3n}$, which contains the (X, Y, Z) coordinates of its n vertices. PCA analysis is applied to all the training 3D shapes. Let $\bar{\mathbf{S}}_3$ be the average 3D shape, $\mathbf{P}_3 \in \mathbf{R}^{3n \times m}$ be the matrix of the first m eigenvectors (in descending order according to their eigenvalues). Any 3D face shape \mathbf{S}_3 can be expressed as

$$\mathbf{S}_3 = \bar{\mathbf{S}}_3 + \mathbf{P}_3 \alpha \quad (1)$$

where $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)^T \in \mathbf{R}^m$ is the shape parameter. $\bar{\mathbf{S}}_3$ is shown in Figure 1.

2.2. 2D Morphable Shape Model

Since shapes in the image are of 2 dimension. To reconstruct the 3D shape, we induce 2D morphable shape models by projection from the 3D morphable shape model in Equation (1) [6, 8].

Suppose that t 2D facial feature points are selected in the images for 3D reconstruction. t vertices, corresponding

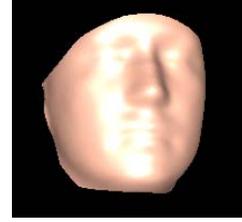


Figure 1. 3D Morphable Shape Model

to the feature points, are also chosen on the face geometry. Let \mathbf{S}_2 be the induced 2D shape. \mathbf{S}_2 can be built in different ways. It can be any combination of the two of the three coordinates. For example, $\mathbf{S}_2 = (X_1, Y_1, \dots, X_t, Y_t)^T \in \mathbf{R}^{2t}$ represents to 2D projection of the 3D shape in the Z -coordinate direction. Thus, \mathbf{S}_2 is the sub shape-vector of \mathbf{S}_3 . According to Equation (1), a 2D face shape \mathbf{S}_2 , assumed zero centered, can be expressed as

$$\mathbf{S}_2 = \bar{\mathbf{S}}_2 + \mathbf{P}_2 \alpha \quad (2)$$

where $\mathbf{S}_2 \in \mathbf{R}^{2t}$ and $\mathbf{P}_2 \in \mathbf{R}^{2t \times m}$ are the sub-coordinates of the feature vertices on \mathbf{S}_3 and \mathbf{P}_3 respectively.

To match the induced 2D shape \mathbf{S}_2 with the 2D face shape in the image \mathbf{S}_2^I , similarity transform is applied. \mathbf{S}_2^I can be obtained from manual labelling, from the face alignment [3, 2] result or from the 3D estimate of Section 3. \mathbf{S}_2 and \mathbf{S}_2^I are matched according to similarity transformation:

$$S_2 = c\mathbf{S}_2^I + T \quad (3)$$

where $T \in \mathbf{R}^{2t}$ is the translation vector and $c \in \mathbf{R}$ is the scale coefficient. Since \mathbf{P}_2 is an orthogonal matrix, α can be derived from Equation (2) as

$$\alpha = (\mathbf{P}_2^T \mathbf{P}_2)^{-1} \mathbf{P}_2 (\mathbf{S}_2 - \bar{\mathbf{S}}_2) \quad (4)$$

To avoid the outliers, the priors are applied to constrain α , and Equation (4) is changed to

$$\alpha = (\mathbf{P}_2^T \mathbf{P}_2 + \lambda \Lambda)^{-1} \mathbf{P}_2 (\mathbf{S}_2 - \bar{\mathbf{S}}_2) \quad (5)$$

where λ is the weighting factor, $\Lambda = \text{diag}(v_1, v_2, \dots, v_m)$ and v_i is the i^{th} eigenvalue.

2.3. Shape Parameter Estimation

As the shape parameters of the 2D and 3D morphable shape models are the same, the 3D shape can be recovered by recovering the 2D shape [6, 8].

To recover the 2D shape, five parameters $\alpha, \mathbf{S}_2, \mathbf{S}_2^I, T, c$ should be estimated. These parameters are iteratively estimated. The estimation works as follows:

- (1) Initialize \mathbf{S}_2 with $\bar{\mathbf{S}}_2$.

- (2) Compute \mathbf{S}_2 with current shape parameter α by Equation (2)
- (3) Estimate the transformation parameters: let T_x and T_y be the average offsets of all t feature points of \mathbf{S}_2^I to the origin along XY axes, respectively, then

$$(T_x, T_y)^T = \frac{1}{t} \sum_{i=1}^t \mathbf{S}_{2i}^I \quad (6)$$

Then $T = (T_x, T_y, \dots, T_x, T_y)$ and

$$c = \frac{\sum_{i=1}^t \langle \mathbf{S}_2^I - (T_x, T_y)^T, \mathbf{S}_2 \rangle}{\sum_{i=1}^t \|\mathbf{S}_2\|^2} \quad (7)$$

- (4) Transform \mathbf{S}_2^I to a new \mathbf{S}_2 : update \mathbf{S}_2 with transformation parameters c and T by Equation (3).
- (5) Estimate new shape parameter α : update α with new \mathbf{S}_2 by Equation (5).
- (6) Go to step (2) if the parameters do not converge.

3. Multi-Image Pose Recovery

For each input image, the pose parameters are estimated as did by Pighin *et al.* [14]. To estimate the pose parameters, an iterative optimization method is used to minimize the difference between the predicted and the observed feature point positions. During the pose recovery process, the positions of the feature positions are also estimated.

3.1. Pose Modelling

To formulate the pose recovery problem, we associate a rotation matrix \mathbf{R}^k and a translation vector \mathbf{t}^k with each camera pose k . The three rows of \mathbf{R}^k are r_x^k , r_y^k , and r_z^k , and the three entries in \mathbf{t}^k are t_x^k , t_y^k , and t_z^k . We write each 3D feature point as \mathbf{p}_i , and its 2D screen coordinates in the k -th image as (x_i^k, y_i^k) . Assuming that the origin of the (x, y) image coordinate system lies at the optical center of each image (*i.e.*, where the optical axis intersects the image plane), the traditional 3D projection equation for a camera with a focal length f^k (expressed in pixels) can be written as

$$x_i^k = f^k \frac{\mathbf{r}_x^k \cdot \mathbf{p}_i + t_x^k}{\mathbf{r}_z^k \cdot \mathbf{p}_i + t_z^k}, \quad y_i^k = f^k \frac{\mathbf{r}_y^k \cdot \mathbf{p}_i + t_y^k}{\mathbf{r}_z^k \cdot \mathbf{p}_i + t_z^k} \quad (8)$$

Instead of using Equation (8) directly, we reformulate the problem to estimate inverse distances to the object [17]. Let $\eta^k = 1/t_z^k$ be this inverse distance and $s^k = f^k \eta^k$ be a world-to-image scale factor. The advantage of this formulation is that the scale factor s^k can be reliably estimated

even when the focal length is long, whereas the original formulation has a strong coupling between the f^k and t_z^k parameters.

Performing these substitution, we obtain

$$x_i^k = s^k \frac{\mathbf{r}_x^k \cdot \mathbf{p}_i + t_x^k}{\eta^k \mathbf{r}_z^k \cdot \mathbf{p}_i + 1}, \quad y_i^k = s^k \frac{\mathbf{r}_y^k \cdot \mathbf{p}_i + t_y^k}{\eta^k \mathbf{r}_z^k \cdot \mathbf{p}_i + 1} \quad (9)$$

If we let $w_i^k = (1 + \eta^k \mathbf{r}_z^k \cdot \mathbf{p}_i)^{-1}$ be the inverse denominator, and collect terms on the left hand side, we get

$$\begin{aligned} w_i^k (x_i^k + x_i^k \eta^k \mathbf{r}_x^k \cdot \mathbf{p}_i - s^k (\mathbf{r}_x^k \cdot \mathbf{p}_i + t_x^k)) &= 0 \\ w_i^k (y_i^k + y_i^k \eta^k \mathbf{r}_y^k \cdot \mathbf{p}_i - s^k (\mathbf{r}_y^k \cdot \mathbf{p}_i + t_y^k)) &= 0 \end{aligned} \quad (10)$$

3.2. Pose Parameter Estimation

Note that these equations are linear in each of the unknowns that we wish to recover, *i.e.*, \mathbf{p}_i , t_x^k , t_y^k , η^k , s^k , and \mathbf{R}^k , if we ignore the variation of w_i^k with respect to these parameters. When solving these equations, we consider that the scalars w_i^k are constant and update their values as better estimates of \mathbf{p}_i , t_x^k , t_y^k , η^k , s^k , and \mathbf{R}^k become available. Using this method, the set of Equation (10) defines a weighted linear least squares problem.

In [14], the parameters are solved in five steps: first s^k , then \mathbf{p}_i , \mathbf{R}^k , t_x^k , t_y^k and finally η^k . For each parameter or set of parameters chosen, the unknowns are solved using linear least squares. However, this process is not very stable as they are not easily to be initialized. To improve the stability, we proposed a four-step method

- (1) Initialize \mathbf{R}^k and s^k . \mathbf{R}^k can be initialized by roughly estimating the rotation of the 3D face model. Of the rest parameters, s^k is the easiest one to initialize. We propose to use the average between-point distance ratio to initialize s^k . According to Equation (8), we have

$$x_i^k = s'^k (\mathbf{r}_x^k \cdot \mathbf{p}_i + t_x^k), \quad y_i^k = s'^k (\mathbf{r}_y^k \cdot \mathbf{p}_i + t_y^k) \quad (11)$$

where

$$s'^k = \frac{f^k}{\mathbf{r}_z^k \cdot \mathbf{p}_i + t_z^k} \quad (12)$$

For any two points i and j , we have

$$x_i^k - x_j^k = s'^k (\mathbf{r}_x^k \cdot \mathbf{p}_i - \mathbf{r}_x^k \cdot \mathbf{p}_j) \quad (13)$$

$$y_i^k - y_j^k = s'^k (\mathbf{r}_y^k \cdot \mathbf{p}_i - \mathbf{r}_y^k \cdot \mathbf{p}_j) \quad (14)$$

So

$$\begin{aligned} \sqrt{(x_i^k - x_j^k)^2 + (y_i^k - y_j^k)^2} = \\ s'^k \cdot \sqrt{(\mathbf{r}_x^k \cdot \mathbf{p}_i - \mathbf{r}_x^k \cdot \mathbf{p}_j)^2 + (\mathbf{r}_y^k \cdot \mathbf{p}_i - \mathbf{r}_y^k \cdot \mathbf{p}_j)^2} \end{aligned} \quad (15)$$

i.e.

$$s'^k = \frac{\sqrt{(x_i^k - x_j^k)^2 + (y_i^k - y_j^k)^2}}{\sqrt{(\mathbf{r}_x^k \cdot \mathbf{p}_i - \mathbf{r}_x^k \cdot \mathbf{p}_j)^2 + (\mathbf{r}_y^k \cdot \mathbf{p}_i - \mathbf{r}_y^k \cdot \mathbf{p}_j)^2}} \quad (16)$$

According to Equation (12), if $\mathbf{r}_z^k \cdot \mathbf{p}_i \ll t_z^k$ (this can be achieved by scaling the image or the 3D model), $s'^k \approx s^k$. So,

$$s^k \approx \frac{\sqrt{(x_i^k - x_j^k)^2 + (y_i^k - y_j^k)^2}}{\sqrt{(\mathbf{r}_x^k \cdot \mathbf{p}_i - \mathbf{r}_x^k \cdot \mathbf{p}_j)^2 + (\mathbf{r}_y^k \cdot \mathbf{p}_i - \mathbf{r}_y^k \cdot \mathbf{p}_j)^2}} \quad (17)$$

where

$$\sqrt{(x_i^k - x_j^k)^2 + (y_i^k - y_j^k)^2} \quad (18)$$

is the distance between point i and j in the image, and

$$\sqrt{(\mathbf{r}_x^k \cdot \mathbf{p}_i - \mathbf{r}_x^k \cdot \mathbf{p}_j)^2 + (\mathbf{r}_y^k \cdot \mathbf{p}_i - \mathbf{r}_y^k \cdot \mathbf{p}_j)^2} \quad (19)$$

is the distance between point i and j of the 3D model in the XY-coordinate. So s^k can be initialized with the average distance ratio of all the point-pairs.

- (2) Solve for (t_x^k, t_y^k, η^k) and s^k iteratively. Given the initialization of \mathbf{R}^k and s^k , (t_x^k, t_y^k, η^k) are linear in Equation (10) and they can be solved together. This is better than solving them independently as there is no initialization needed for them. (t_x^k, t_y^k, η^k) and s^k iteratively solved until they converge.
- (3) Solve for \mathbf{R}^k , (t_x^k, t_y^k, η^k) and s^k iteratively. With the converged result of step (2), \mathbf{R}^k can be solved better.
- (4) Solve for \mathbf{R}^k , (t_x^k, t_y^k, η^k) , s^k , and \mathbf{p}_i iteratively.

4. Multi-Image Texture Extraction

The morphable shape model in Section 2 and the pose recovery in Section 3 can only give the shape of the 3D face. The texture should also be extracted. Unlike the texture extraction in [6, 8] which only use only image, we adopt the multi-image texture extraction in [14] to get more realistic face texture.

The multi-image texture extraction problem can be defined as follows. Given a collection of photographs, the recovered viewing parameters, and the 3D face model, compute for each point \mathbf{p} on the face model its texture color $T(\mathbf{p})$. The texture value $T(\mathbf{p})$ at each point on the face model can be expressed as a convex combination of the corresponding colors in the photographs:

$$T(\mathbf{p}) = \frac{\sum_k m^k(\mathbf{p}) I^k(x^k, y^k)}{\sum_k m^k(\mathbf{p})} \quad (20)$$

Here, I^k is the image function (color at each pixel of the k -th photograph), and (x^k, y^k) are the image coordinates of the projection of \mathbf{p} onto the k -th image plane. The weight map $m^k(\mathbf{p})$ is a function that specifies the contribution of the k -th photograph to the texture at each facial surface point. There are several important considerations that must be taken into account when defining a weight map:

- Self-occlusion: $m^k(\mathbf{p})$ should be zero unless \mathbf{p} is front-facing with respect to the k -th image and visible in it.
- Smoothness: the weight map should vary smoothly, in order to ensure a seamless blend between different input images.
- Positional certainty: $m^k(\mathbf{p})$ should depend on the “positional certainty” [9] of \mathbf{p} with respect to the k -th image. The positional certainty is defined as the dot product between the surface normal at \mathbf{p} and the k -th direction of projection.

In order to support rapid display of the textured face model from any viewpoint, it is desirable to blend the individual photographs together into a single texture map. This texture map is constructed on a virtual cylinder enclosing the face model. The mapping between the 3D coordinates on the face mesh and the 2D texture space is defined using a cylindrical projection, as in several previous papers [9, 11].

We will index the weight map m^k by the (u, v) coordinates of the texture being created. Each weight $m^k(u, v)$ is determined by the following steps:

1. Construct a visibility map F^k for each image k . These maps are defined in the same cylindrical coordinates as the texture map. We initially set $F^k(u, v)$ to 1 if the corresponding facial point \mathbf{p} is visible in the k -th image, and to 0 otherwise.
2. Compute the 3D point \mathbf{p} on the surface of the face mesh whose cylindrical projection is (u, v) . This computation is performed by casting a ray from (u, v) on the cylinder towards the cylinder's axis. The first intersection between this ray and the face mesh is the point \mathbf{p} . Let $P^k(\mathbf{p})$ be the positional certainty of \mathbf{p} with respect to the k -th image.
3. Set weight $m^k(u, v)$ to the product $F^k(u, v)P^k(\mathbf{p})$.

5. Morphable Reconstruction with Multiple Images

With multiple images, the coordinates of the feature points \mathbf{p}_i can be estimated. Pighin *et al.* [14] estimated the coordinates of other face points by interpolation. However,

the estimation is not accurate enough so that shape refinement with much more additional correspondence, *e.g.* 99 additional correspondence used in [14]. Another problem is that if there are some errors with the feature points estimation, there is no way to correct them. These problems can be solved by morphable models. With morphable models, we do not need to specify additional correspondence as the coordinates of other points can be estimated by the morphable model, and the error can also possibly be corrected by the morphable model.

However, as one 2D morphable model can only estimate 2 coordinates, we propose to use two orthogonal 2D morphable models to utilize the multiple images: frontal and profile morphable models.

The frontal 2D morphable model is:

$$\mathbf{S}_f = \bar{\mathbf{S}}_f + \mathbf{P}_f \alpha \quad (21)$$

where $\mathbf{S}_f = (X_1, Y_1, \dots, X_t, Y_t)$, $\bar{\mathbf{S}}_f$, and \mathbf{P}_f comes from the (X, Y) coordinates of the 3D shape \mathbf{S}_3 , the 3D mean shape $\bar{\mathbf{S}}_3$ and the 3D eigen matrix \mathbf{P}_3 .

The profile 2D morphable model is:

$$\mathbf{S}_p = \bar{\mathbf{S}}_p + \mathbf{P}_p \alpha \quad (22)$$

where $\mathbf{S}_p = (Y_1, Z_1, \dots, Y_t, Z_t)$, $\bar{\mathbf{S}}_p$, and \mathbf{P}_p comes from the (Y, Z) coordinates of the 3D shape \mathbf{S}_3 , the 3D mean shape $\bar{\mathbf{S}}_3$ and the 3D eigen matrix \mathbf{P}_3 .

The process of the 3D reconstruction works in an iterative way:

- (1) Estimate the coordinates of the feature points \mathbf{p}_i with the pose recovery method in Section 3. If the coordinates converge with the previous estimation, go to step (4).
- (2) Update the 3D model with shape parameter estimation from the frontal and profile morphable models.
 - (2.1) Estimate shape parameter with frontal model.
 - (2.2) Estimate shape parameter with profile model.
 - (2.3) If not converge, go to step (2.1).
- (3) Go to step (1) to estimate the coordinates of the feature points \mathbf{p}_i with the new 3D model.
- (4) Extract the texture by the texture extraction method in Section 4.

6. Experiments

We used face images in PIE [16] with different poses to reconstruct the 3D faces. Feature points, including eye corners, nose tip, mouth corners and ear corners, are manually

labelled on the face images. Figure 2 shows the experimental results for 3 persons. Face images in the left column are the input faces for 3D reconstruction, three faces for each person. The middle column contains the 3D reconstruction results with the proposed method. And the right column shows the 3D reconstruction results with only one frontal image. These results show that our proposed method is quite effective and the 3D reconstruction results with it are very realistic. It performs much better than 3D reconstruction only with one frontal image, especially for the side view. The whole algorithm runs for 3 to 4 seconds for one image on a P4-1.8GHz PC with 512RAM. It is tens of times faster than the 3D morphable model [15]. The speed can be even improved by reducing the vertices of the morphable shape model.

7. Conclusion

We have proposed an efficient 3D face reconstruction method by morphable model with multiple images. This method combines the advantages of both the morphable model and multi-image reconstruction. With multiple images, the 3D coordinates of the feature points can be estimated. And frontal and profile 2D morphable models are proposed to estimate the 3D morphable model. The 3D face is iteratively estimated by the above two steps. Finally, the texture is extracted from multiple images with the pose estimation from the estimated 3D face. The experimental results are very effective and promising. Future work includes applying the 3D model to face animation and recognition, and using robust multi-view face alignment to automate the reconstruction.

References

- [1] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH'99 Conference Proceedings*, pages 187–194, 1999.
- [2] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *ECCV98*, volume 2, pages 484–498, 1998.
- [3] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models: Their training and application. *CVGIP: Image Understanding*, 61:38–59, 1995.
- [4] D. DeCarlos, D. Metaxas, and M. Stone. An anthropometric face model using variational techniques. In *SIGGRAPH'98 Conference Proceedings*, pages 67–74. ACM, 1998.
- [5] S. DiPaola. Extending the range of facial types. *Journal of Visualization and Computer Animation*, 2(4):129–131, 1991.
- [6] Y. X. Hu, D. L. Jiang, S. C. Yan, L. Zhang, and H. Zhang. Automatic 3d reconstruction for face recognition. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pages 843–848, 2004.

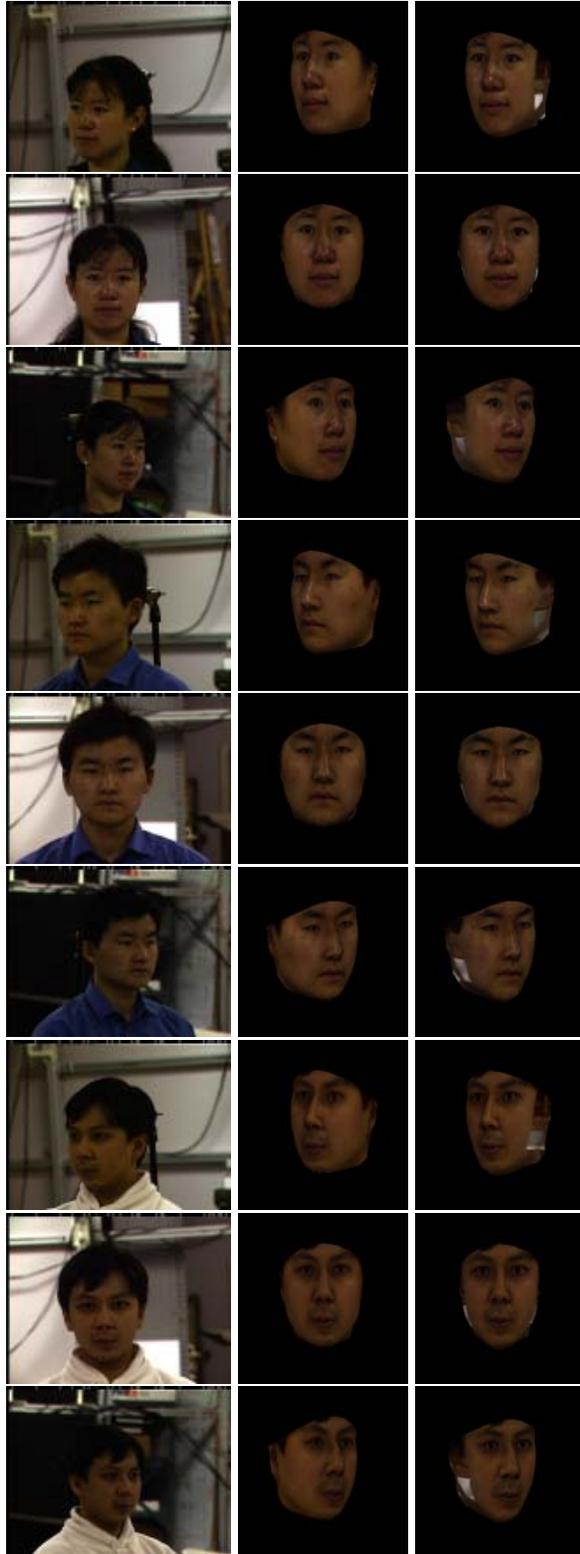


Figure 2. 3D Reconstruction from 3 faces. Left column: input faces; Middle column: results with multiple images; Right column: results with single image.

[7] H. H. S. Ip and L.-J. Yin. Constructing a 3d head individualized model from two orthogonal views. *The Visual Computer*, 12(5):254–266, 1996.

[8] D. Jiang, Y. Hu, S. Yan, L. Zhang, H. Zhang, and W. Gao. Efficient 3d reconstruction for face recognition. *Pattern Recognition*, 38:787–798, 2005.

[9] T. Kurihara and K. Arai. A transformation method for modeling and animation of the human face from photographs. In N. M. Thalmann and D. Thalmann, editors, *Computer Animation*, pages 45–58, Tokyo, 1991. Springer-Verlag.

[10] K.-M. Lam and H. Yan. An analytic-to-holistic approach for face recognition based on a single frontal view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(7):673–686, 1998.

[11] Y. Lee, D. Terzopoulos, and K. Waters. Realistic modeling for facial animation. In *SIGGRAPH'95 Conference Proceedings*, pages 55–62. ACM SIGGRAPH, August 1995.

[12] Z. Liu, Z. Zhang, C. Jacobs, , and M. Cohen. Rapid modeling of animated faces from video. In *The 3rd International Conference on Visual Computing*, pages 58–67, Mexico City, 2000.

[13] F. Parke. Computer generated animation of faces. In *ACM National Conference*. ACM, November 1972.

[14] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. Salesin. Synthesizing realistic facial expressions from photographs. In *SIGGRAPH'98 Conference Proceedings*, pages 75–84. ACM, July 1998.

[15] S. Romdhani, V. Blanz, and T. Vetter. Face identification by fitting a 3d morphable model using linear shape and texture error functions. In *Proceedings of the European Conference on Computer Vision*, volume 4, pages 3–19, 2002.

[16] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression (pie) database. In *The 2002 International Conference on Automatic Face and Gesture Recognition*, 2002.

[17] R. Szeliski and S. B. Kang. Recovering 3d shape and motion from image streams using nonlinear least squares. *Journal of Visual Communication and Image Representation*, 5(1):10–28, March 1994.

[18] R. Zhang, P. S. Tai, J. E. Cryer, and M. Sha. Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, 1999.