Automatic 3D Face Texture Mapping Framework from Single Image

X.C. He, S.C. Yuk, K.P. Chow, K.-Y. K. Wong, R. H. Y. Chung
The Department of Computer Science, the University of Hong Kong, Pokfulam Road, Hong Kong
{xche, scyuk, chow, kykwong, hychung}@cs.hku.hk

ABSTRACT
This paper proposes a novel face texture mapping framework for 3D face reconstruction from a single frontal view or half-profile view facial image. Face reconstruction method that originates from the proposed framework, unlike most of the existing ones, is novel in the sense that it is not tightly coupled to a specific face model, and yet it simplifies the pose estimation problem which is pivotal to the success of face reconstruction. This paper details the proposed framework, and illustrates how it addresses the ill-posed pose estimation problem, of which the solution is optimal in the least square sense. With accurate pose estimation of face, precise texture mapping is thus made possible to allow photo realistic rendering of face images in the 3D space. Experimental results demonstrates that reliable and photo realistic 3D face reconstruction can be easily realized in our framework by utilizing a generic 3D face model, standard Haar-like feature based detector and active appearance model. With proposed frame work, face recognition systems could be more robust to pose changes by reconstructing frontal faces from non-frontal ones.

Keywords
Face reconstruction; Pose estimation; Texture mapping; Face model

1. INTRODUCTION
Face detection and recognition algorithms have been very hot research topics in the last decades and a lot of robust and practical algorithms have been proposed and developed. Recently, these algorithms have become very mature and reliable to the extent that they have already been put into practical applications for use in industrial and consumer electronics. However, the great success of these technologies rests on a very restrictive assumption, in which the face of interest is normally assumed to have an orientation that is passport-photo alike. As a result, though it seems that technological speaking face detection and recognition algorithms are mature, it is hard to adopt these technologies for use in practical environments due to the fact that pictures of human faces are not necessarily passport-photo alike.

ICIMCS'09, November 23–25, 2009, Kunming, Yunnan, China.

To overcome these hurdles, face reconstruction is normally required so that human face can be reconstructed and transformed to a form that is suitable for use by a particular application, and several strategies have been explored in previous works to address this problem. Our literature review reveals that previous face reconstruction technologies either require manual assistance [1], or long-winded iteration procedure [2]. Moreover, they usually need a frontal image [3] or multi-view images [4] as input. In view of this, a rapid and automatic 3D face texture mapping framework from a single non-frontal view image is proposed in this paper. The most related work to this paper is proposed by Park et al. [5]. They proposed a method to reconstruct a 3D face from a single 2D image based on pose estimation and a deformable model of 3D face shape, and expectation maximization algorithm is adopted in pose estimation. Our work is different from [5] in several aspects:

1) Texture maps are included in our face model instead of one-color-per-vertex mechanism. By using texture mapping technology, detailed texture could be rendered on a shape or polygon with fewer vertices.

2) In our pose estimation procedure, 3D-2D transformation matrix is estimated by calculating pseudo inverse matrix of 3D coordinates, which is actually a least square solution to the 3D to 2D fitting problem. When compared with expectation maximization algorithm in [5], our approach is more simple and convenient, since estimation of pose parameters like rotation, scaling and translation are not necessary.

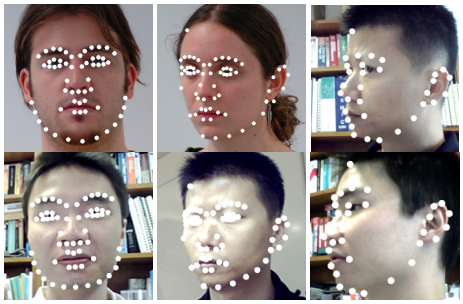
3) In [5], 79 detected feature points were triangulated, and texture of each triangle in 2D image were interpolated into the corresponding triangle in 3D face, based on the assumption that points in each triangle are on a plane. With only 79 feature points, triangles could be quite large, topographic variation in triangle of 3D face could introduce defects on the reconstructed result. In contrast, our 3D mesh is projected on 2D images according to the result of pose estimation, and the texture of each patch in 2D image are interpolated into corresponding patch in texture map. Since our texture mapping framework is not tightly coupled to a particular mesh model, which allows us to use a generic and dense 3D mesh, and thus each facet could be regarded as in a plane, or even a texel in texture mapping.

Basically, the proposed framework includes several parts, which are facial feature detection, 3D face modeling, pose estimation, and texture mapping. Details of each step will be discussed in Section 2 to Section 5. Face reconstruction experimental results are presented in Section 6 to illustrate the efficiency of the proposed algorithm. Section 7 concludes the whole paper.

2. FACIAL FEATURE DETECTION

The very first step of this framework is to detect and classify the input human face as frontal, half-profile or profile face. To illustrate the effectiveness of the proposed texture mapping framework, we deliberately selected a well known face detector together with active appearance model in realizing our face reconstruction method. We expect the performance of the proposed face reconstruction would be at least equal, if not better, when more sophisticated feature detectors are employed. Essentially, the popular face detector [6], namely a cascade of boosted classifiers working with Haar-like features, is adopted. It should be noted that we need two detectors here, one for frontal face, and the other for profile face. Typically profile face detector can only detect face towards one direction, and face to another direction can be detected by simply flipping input image horizontally. Without loss of generality, we just assume all the profile faces are oriented towards the left side.

Then, by using the active appearance model (AAM) algorithm [7] we can match the pre-trained face appearance model to the input image and locate the detailed facial features. Similar to the method in [4], we use three distinct appearance models, which are frontal, half-profile and profile. The training set consists of labeled images, where key landmark points are manually marked on each example face. Figure 1 shows examples of labeled images used to train the three distinct models.



Frontal **Half-Profile** **Profile**

Figure 1. Examples from the training sets for AAM

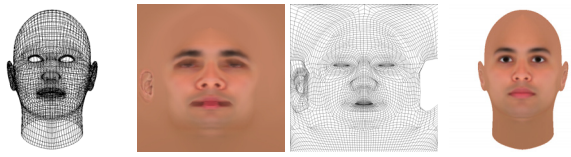
The idea of AAM search is to minimize the difference between an input image and the one synthesized by the appearance model. The difference could be defined as the magnitude of difference vector:

$$\Delta \mathbf{I} = \left| \mathbf{I}_i - \mathbf{I}_m \right|^2 \quad (1)$$

where $\mathbf{I}_i$ is the vector of grey-level values in the image, and $\mathbf{I}_m$ is the vector of grey-level values for the current model parameters. Individual models are then applied to match the input face image and search for best fit. The one with minimum value of $\Delta\mathbf{I}$ is adopted for locating the facial features. AAM algorithm includes an initialization procedure and a search procedure. During initialization, scale and offset of model will be coarsely determined, where the accuracy of search will depend on it. Since Haar-feature face detector is faster than AAM initialization, it will be more efficient to use result of Haar-feature face detector as input of AAM algorithm to reduce the dynamic range of the subsequent search space.

3. THREE DIMENSION FACE MODELING

In some reference [2, 5, 8], a 3D face was represented by a shape-vector S and a texture-vector T, where shape-vector contains 3D coordinates and texture-vector contains the R, G, B color values of all the vertices. Since there is only one color value per vertex, to achieve a high resolution appearance, the 3D mesh of face model should contain a large number of vertices (approximately 70,000 vertices in [2]). In this proposed framework, we employ texture mapping technology instead, where detailed texture could be rendered on a 3D shape with substantially fewer vertices. To illustrate the effectiveness of the framework, the face model we employed in this paper comes from a 3D mesh with less than one-tenth of the vertexes (6292 vertices and 6152 facets) of the mesh employed in [2]. The face model, as well as the texture maps and texture coordinates are depicted in Figure 2(a)-(c). In summary, texture map is a 2D image which contains all the texture of a human face in some sort. Texture coordinates associate a particular location in the texture map with vertices in 3D mesh. These coordinates determine how the texture is mapped from the 2D texture map onto the final 3D mesh shown in Figure 2(d). It should be noted that this face model is derived from FaceGen® (a software product of Singular Inversions Inc.), where the 3D mesh contains several objects including skin, eyes, sock, teeth and tongue, which is useful for morphing 3D face into a variety of face expressions. Texture map of skin is shown in Figure 2(b), and other texture maps are omitted here. In our work, the texture maps of skin and eyes will be rebuilt according to the input image, while default texture maps will be employed for the rest.



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

Figure 2. Three-D face model. (a) 3D mesh, (b) Skin texture map, (c) Texture coordinates map, (d) Texture rendered

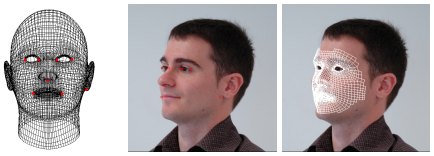
It is well known that 3D face reconstruction includes shape reconstruction and texture reconstruction. This paper focused on the latter, and therefore a single mean shape was adopted in this work. However, 3D face geometry reconstruction algorithm in [8] could be utilized to get a personalized 3D shape if required. Detected facial features in Section 2 could be utilized for 2D face alignment then.

4. POSE ESTIMATION

In order to extract texture of input image for filling patches in 3D mesh, coordinates of each mesh points has to be projected onto the input image. Therefore we need to estimate the pose of the 3D face to align the mesh points properly with the 2D face in the input image. Fortunately, by using 2D coordinates of facial features detected by AAM in the input image and their corresponding 3D coordinates in 3D mesh as a hint, the transformation matrix for this projection can be deduced as follows:

Let $\mathbf{Q}=(X_Q, Y_Q, Z_Q)$ be an arbitrary point in 3D, let $\mathbf{q}=(x_q, y_q)$ be the corresponding 2D image coordinates of **Q**. A forward

mapping function, Φ, which defines the transform function from a point in 3D coordinates to a point in the 2D image coordinates is given as, $\mathbf{q} = \Phi\{\mathbf{Q}\}$.



(a) **(b)** **(c)**

Figure 3. (a) Feature points on 3D mesh, (b) Feature points in 2D input image, (c) Fit 3D mesh into 2D image.

By perspective transformation, we have

$$\begin{bmatrix} x_q & y_q & z_q & 1 \end{bmatrix} = \begin{bmatrix} X_Q & Y_Q & Z_Q & 1 \end{bmatrix} \cdot \mathbf{T} \quad , \qquad (2)$$

where $\mathbf{T}$ is the transformation matrix. Then we assume that sub-matrix $\mathbf{T}_1$ is the first two column of $\mathbf{T}$, we have

$$\begin{bmatrix} x_q & y_q \end{bmatrix} = \begin{bmatrix} X_Q & Y_Q & Z_Q & 1 \end{bmatrix} \cdot \mathbf{T}_1 \quad . \qquad (3)$$

Among all the detected facial features, we pick some for pose estimation, e.g. eye corners, mouth corners, ear tips and nose tip, as shown in Figure 3 (a)-(b). Note that our framework is less tightly coupled to the 3D model employed, in the sense that virtually any 3D face model can be employed as long as there exists a direct correspondence between key facial feature points and those in the mesh. Assume *m* feature points are utilized, we have

$$\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ . & . \\ x_m & y_m \end{bmatrix} = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ X_2 & Y_2 & Z_2 & 1 \\ . & . & . & 1 \\ X_m & Y_m & Z_m & 1 \end{bmatrix} \cdot \mathbf{T}_1 \qquad (4)$$

Then $\mathbf{T}_1$ can be calculated as

$$\mathbf{T}_1 = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ X_2 & Y_2 & Z_2 & 1 \\ . & . & . & 1 \\ X_m & Y_m & Z_m & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ . & . \\ x_m & y_m \end{bmatrix} \qquad (5)$$

where $[\cdot]^{-1}$ represent Moore-Penrose pseudo inverse. The matrix thus obtained is actually a least square solution of the mesh fitting problem. With this matrix, projection of every vertex of 3D mesh onto the input image could be calculated by Eqt.(3), which enables us to rebuild the texture map.

5. REBUILDING TEXTURE MAP

After pose estimation, texture map could be rebuilt by filling each texture element with those extracted from the input image. It should be noted that only central region of texture map, which includes the very feature of human face, is needed to be rebuilt. Coordinates of mesh in central face projecting on the input image are calculated and marked as shown in Figure 3(c), where every patch has a correspondence in the texture coordinates map. For an individual patch of mesh, we just map the texture of this patch in the input image into the corresponding patch in the texture map. Since this mapping is from 2D image to 2D image, affine transform could be utilized for interpolation. Affine transform is determined by the coordinates of all vertices of this patch in the two images. The transformation and the filling procedure are illustrated in Figure 4(a).

Central region of texture map (called face region) is filled with the texture that comes from the input image, while the other parts of it (called non-face regions) are filled with default texture. A post processing is performed to smooth out the transition values between these two regions for better 3D face rendering results. First, color of non-face region is modified to that of the color tune of the face region. Then boundary of two regions is blurred to smooth out the color transition from one region to the other. On the other hand, for non-frontal face, the texture of the occluded face region cannot be reconstructed well as they are not visible in the 2D input image. Therefore, the post processing step also includes a symmetrization process to fill visible texture into occluded region and to make the texture map lateral symmetry. These post processing steps are illustrated in Figure 4(b).

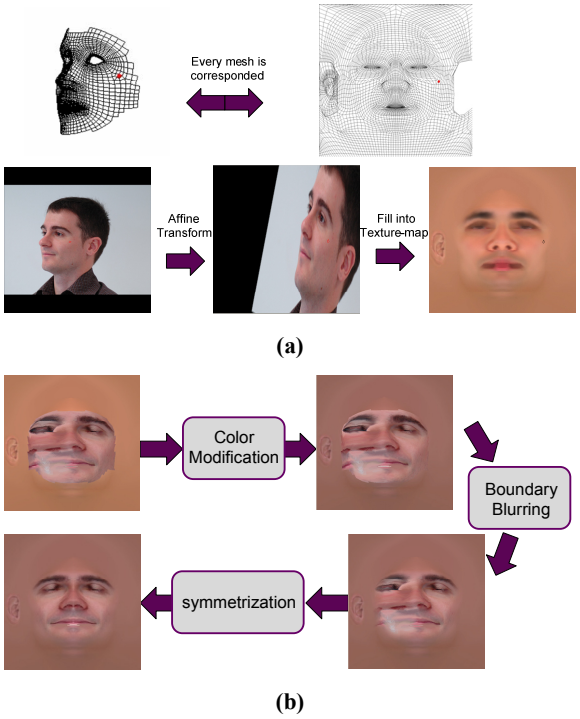
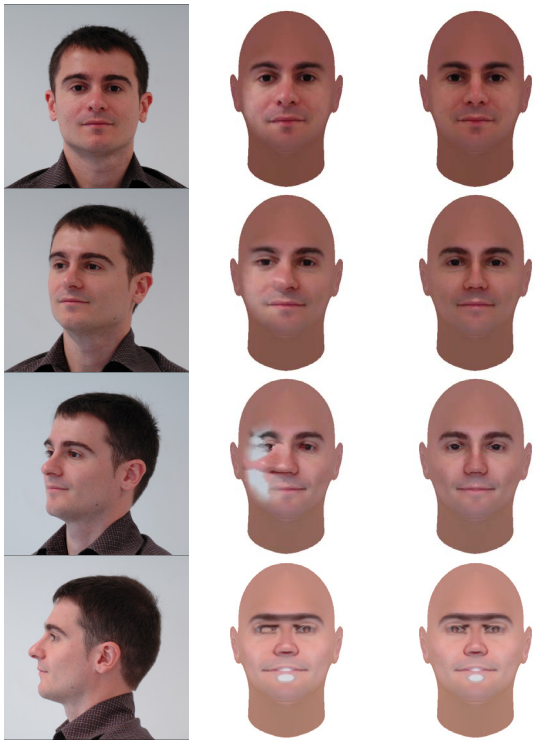


Figure 4. Diagram of rebuilding skin texture map. (a) Transform and filling. (b) Post processing.

It was mentioned before that our face model includes several objects and texture maps. The above process illustrates how skin texture map can be rebuilt. It should be noted that eyes texture map could also be rebuilt in a similar fashion.

6. EXPERIMENTS AND DISCUSSIONS

GTAV face database [9] is adopted in our experiments to serve as a common reference for performance evaluation. Furthermore, a



(a) (b) (c)

Figure 5. Reconstruction results with different poses. (a) Original image, (b) Synthetic results without symmetrization, (c) Synthetic results with symmetrization.

(a) (b) (c)

Figure 6. More reconstruction results. (a) Original image, (b) Synthetic results of frontal view, (c) Synthetic results of various poses.

USB web camera is utilized for capturing more face images for testing the robustness of the framework in reconstructing faces from a complex background. Input images were all normalized to a size of 512×512, and the typical processing time are 1~2 seconds, which varies according to the size of the face in the image. Among all the several steps of the proposed framework, facial feature detection consumes most of time.

We test face images with different poses in the GTAV database, and Figure 5 shows some results of it. The images in the first row, Figure 5(a), shows the input 2D images with different poses, including one frontal view, one profile view and two half-profile views. Figure 5(b) shows reconstructed frontal views by the proposed method without symmetrization procedure, and Figure 5(c) shows synthetic frontal views with symmetrisation procedure. The experiment results in Figure 5(b) reveal that the closer to profile view, the worse texture mapping performance we can get from the occluded face. Figure 5(c) demonstrates that the proposed method can rectify this to produce reliable results through the symmetrization process. Except the profile view in the last column, which suffers a little bit due to the lack of texture information in the eyes and mouth regions, the other reconstructed results appear to be reliable and realistic enough for use by subsequent processing step such as face recognition.

More reconstruction results are demonstrated in Figure 6, where the frontal view results are shown in Figure 6(b), and Figure 6(c) shows synthetic faces with various poses. When the face in the input image is a non-frontal view (first three columns in Figure 6), symmetrization processing will be performed, which will be determined automatically according to which active appearance model is adopted.

Furthermore, a preliminary experiment of utilizing FaceVACS® to evaluate validity of this framework is also presented here. FaceVACS® is a face recognition software developed by Cognitec Systems Ltd., which guarantee a true match rate of more than 98% at a typical false match rate of 0.1%, under the condition that the faces must be visible at an almost frontal view position with no more than +/- 15 degrees rotation. In this experiment, two subsets of FERET database [10] were used: a gallery subset of 1000 images containing frontal faces of 1000 persons (typical images are shown in Fig. 7(a)), and a probe subset of 100 images containing non-frontal faces of 100 persons (shown in Fig. 7(b)). Among those 100 images in the probe subset, 50 images contain quarter profile faces with pan rotation of 22.5 degree, and other 50 images contain half profile faces with pan rotation of 67.5 degree. As shown in Table 1, 19 images with quarter profile faces can be correctly identified by the face recognition software; while no images with half profile faces can be identified. We then apply our proposed face reconstruction method on this probe subset, and use those reconstructed frontal faces, as illustrated in Fig. 7(c), for identification. With our face reconstruction in place, an increase from 19 to 43 successful identifications of quarter profile faces is observed, while another

(a) (b) (c)

Figure 7. (a) gallery subset with frontal faces, (b) Probe subset with non-frontal faces, (c) Reconstructed frontal faces from (b).

increase from 0 to 36 successful identifications of half profile faces is also noted. From this preliminary experiment, our proposed framework seems to have good potential in enhancing the robustness of face recognition systems to various face poses.

Table 1. Recognition Results

	Pan rotation	Identified	Could not Identified	Total
Profile faces	22.5	19	31	50
	67.5	0	50	50
Reconstructed frontal faces	22.5	43	7	50
	67.5	36	14	50

7. CONCLUSION AND FUTURE WORK

In this paper, an automatic texture mapping framework from single 2D face image to 3D face is proposed. Texture in 2D image are extracted and then rendered on 3D face directly with high fidelity, instead of optimal fitting of texture from 3D face database. Experimental results illustrates that non-frontal view face can be synthesized to a realistic frontal view, which could be utilized to improve recognition accuracy when dealing with non-frontal face images.

There is a wide variety of applications for 3D face reconstruction from 2D image. Apart from non-frontal to frontal transformation, the results can also be used for automatic post-processing of a face within the original picture or movie sequence. The face can be combined with other 3D graphic objects, such as hairs, hats or glasses, and then rendered in front of special background. Furthermore, we can change the appearance or expression of the face by changing special attributes of 3D models, as shown in Figure 8.

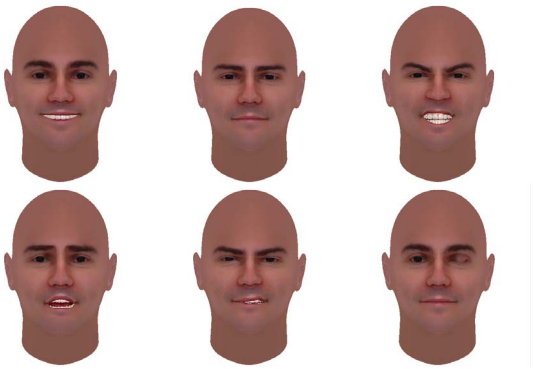


Figure 8. Synthetic results with various expressions.

In future work, we expect to 1). Use a morphable model to fit input face image to personalized 3D shape; 2) Apply the above proposed framework on video, to reconstruct 3D faces with higher resolution from multiple video frames; and 3) Conduct more in-depth experiments to evaluate the performance of proposed method.

8. ACKNOWLEDGEMENTS

This research was sponsored by the Innovation and Technology Commission of the Government of the Hong Kong Special Administrative Region, under the Grant ITS/174/08.

Our special thanks go to Cognitec Systems Ltd., for supporting us to use FaceVACS software to evaluate our proposed method.

9. REFERENCES

[1] K.M. Lam and H. Yan, "An analytic-to-holistic approach for face recognition based on a single frontal view", *IEEE Trans. on PAMI*, vol. 20, pp. 673-686, 1998.

[2] V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model", *IEEE Trans. on PAMI*, vol. 25, pp. 1063-1074, 2003.

[3] Y. Hu, D. Jiang, S. Yan, L. Zhang, and H. Zhang, "Automatic 3D reconstruction for face recognition", *Proc. of IEEE Conf. on AFGR*, pp 843-848, 2004.

[4] T.F. Cootes, K. Walker, and C.J. Taylor, "View-based active appearance models", *Proc. of IEEE AFGR*, pp 227-232, 2000.

[5] S. W. Park, J. Heo, and M. Savvides, "3D face reconstruction from a single 2D face image", presented at *IEEE Conf. on CVPR Workshops*, pp 1-8, 2008.

[6] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection", *Proc. of IEEE ICIP*, vol. 1, pp 900-903, 2002.

[7] T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active appearance models", *IEEE Trans. on PAMI*, vol. 23, pp. 681-685, 2001.

[8] Z. Zhang, Y. Hu, T. Yu, and T. Huang, “Minimum variance estimation of 3D face shape from multi-view”, *Proc. of IEEE Conf. on AFGR*, pp 547-552, 2006.

[9] F. Tarres and A. Rama, “GTAV Face Database”, available at http://gps-tsc.upc.es/GTAV/ResearchAreas/UPCFaceDatabase/GTAVFaceDatabase.htm

[10] "The Facial Recognition Technology (FERET) Database", http://www.itl.nist.gov/iad/humanid/feret/feret_master.html