A FAST 3D FACE RECONSTRUCTION METHOD FROM A SINGLE IMAGE USING ADJUSTABLE MODEL

*Tao Wu , Fei Zhou, Qingmin Liao**

Visual Information Processing Laboratory/Shenzhen Key Laboratory of Science and Technology
Department of Electronic Engineering/Graduate School at Shenzhen, Tsinghua University,China

ABSTRACT

In this paper, we propose a fast and robust method which uses only a single frontal face image as input to reconstruct a plausible 3D face. Our method mainly consists of three stages: feature point detection, model adaptation in X-Y plane and model adjustment on Z-axis direction . At first stage, we detect some face regions such as face contour and facial components automatically. In these regions, we extract several feature points which can generally describe the structure of face. Subsequently, we apply several deformation processes and optimization procedures on an adjustable 3D face model in the X-Y plane based on these feature points. Finally, we present a method of insertion to obtain a dense and smooth model. Experimental results demonstrate the effectiveness and efficiency of our method as well as the robust adaptation to the complex imaging condition.

Index Terms— 3D face reconstruction, adjustable model, model adaption, insertion

1. INTRODUCTION

Three-dimensional(3D) face reconstruction has always been a challenging and difficult task because the geometric structure of face is complex and individualities. The traditional 3D face reconstruction methods are based on multi-perspective, such as [1] [2] [3]. However, these methods need to match the images from different perspectives before reconstructing the 3D face, which reduces the efficiency.

Recently, researchers pay more and more attention to the technologies of 3D face reconstruction from single image because it avoids the image registration problem and has a high value of practical application. Nevertheless, reconstructing 3D face from single image faces many difficulties. The depth information is hard to obtain from a single image. Complex illumination condition and changeful expression also affect the reconstruction result to a large extent. To solve these problems, some preliminary attempts have been made for the past several years, which can be classified as either "shape from X" approaches or learning based techniques. The former methods take advantage of the clues which are caused by external conditions such as illumination [4] [5]. However, these methods are highly restricted by the input images because the motion and illumination conditions are usually uncertain and the depth variations of face are hard to predicted.

The latter methods learn a 3D shape from a single image on the basis of the available 3D face database. Notable methods prefer to learn a model from a mass of training examples [6] [7] [8] [9]. They attempt to learn a face model which can best fit the input images from a large 3D face database. These methods all require carefully aligned 3D face scans to be assembled and then used to learn the space of faces. Inthe-Wild face Reconstruction method was recently proposed by [10], combining an example based approach with a shape from shading method. However, these previous methods are time-consuming and require a large number of reference models, which further limit their applicability.

In this paper, a novel method is proposed to solve the problem of 3D face reconstruction from a single image with neither any training processes nor a large number of training examples. First, several feature points are detected in the input image and match them to the ones in a selected model. Then, we adjust the model on X-Y plane for adapting it to the input image. Finally, we present a method of insertion on each surface in the model to make the reconstruction results more smooth and dense. Experiment results show that our algorithm is both fast and robust, which can be readily used in many applications automatically.

2. PROPOSED METHOD

In this section, we present the whole system of our method about the 3D face reconstruction from a single frontal face image. The following subsections will describe the details.

This work was supported by the National Natural Science Foundation of China under Grant No.61271393 and 61301183, the China Post-Doctoral Science Foundation under Grant No. 2013M540947 and 2014T70083, and the Special Foundation for the Development of Strategic Emerging Industries of Shenzhen under Grant No. JCYJ20140417115840272 and JCYJ20150331151358138.

*Corresponding Author, E-mail: liaoqm@tsinghua.edu.cn

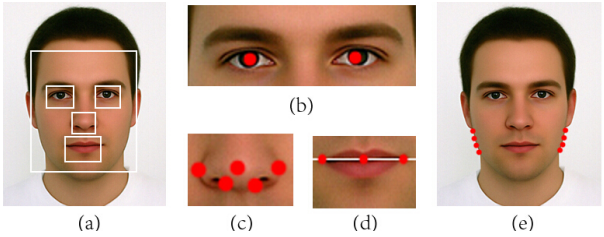


Fig. 1. Feature points detection results. (a) Feature points on face contour. (b) Local regions detection results inside face. (c), (d) and (e) Feature points in different local regions.

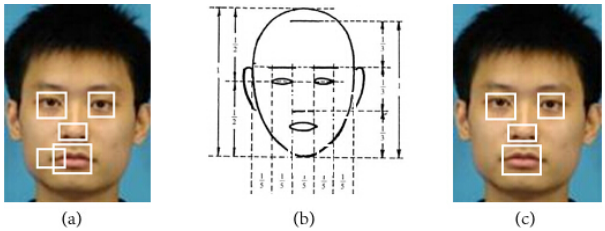


Fig. 2. (a) Uncorrected detection result, (b) Human facial structure, (c) Modified result.

2.1. Feature points detection

The feature points of face can be classified into two categories as those which represent the face components and those which represent the face contour. The first category includes the feature points such as canthus, tips of nose, corners of the mouth. To get the feature points in the second category, we use four standard classifiers in [11] [12] [13] to detect the location of facial features, following the work of [13]. The detection results are shown in Fig.1(a). However, the detection results are not always satisfied. This is mainly because that it is hard to ensure the quality of the input images such as the resolution and noise magnitude. To overcome this problem, we use the common structure of facial features as the restriction to obtain the better detection results. As shown in Fig.2(a), two mouth regions are detected. For each mouth region, we judge whether it is in the mouth region of the face as shown in Fig.2(b) and exclude the incorrect mouth regions. The detection result after corrected by the restriction is shown in Fig.2(c) that the incorrect mouth region in the left mouth corner is excluded. In the region of eyes, we select the geometric center of each region as the feature point which is shown in Fig.1(b). For nose region, we use the geometric center of the region as the feature point of nasal tip. Then harris corner detector [14] is used to detect the feature points of the nose region for getting the position of the nostrils and nose wings. The result is shown in Fig.1(c). For mouth region, we take advantage of transverse projection and corner detector in [14] to detect the centerline and the corners of the mouth. The result is shown in Fig.1(d). To get the feature points in the second category, we use the method based on the color of skin [15] to detect the face region in the input image. To obtain the feature points in the face contour, we sample uniformly in the face edges between the nose tip and the mouth in vertical direction. An example of detected feature points on face contour is shown in Fig.1(e).

2.2. Model processing in X-Y plane

In consideration of the difficulty of accurately adjusting the face contour and the efficiency of the algorithm, we use two adjustable face models shown in Fig.3(a) and (b) to balance the efficiency and effectiveness of our method. The two models represent the sharp face and round face respectively, which can approximately represent most people's facial form. The numbers of vertices and surfaces in each model are denoted as N and M respectively. To select the most suitable model for each of the input images, a simple and effective method is presented. We calculate the slope of the straight line which is fitted by the feature points of the first category on one side of the face contour, as shown in Fig.1(a). If the slope k satisfy $|k| < t$, the former model is selected. Otherwise we select the latter model. t is a threshold which is set as 3.5.

After selecting the model, three steps are taken to adjust the model. First, we normalize the face model to the same scale as the input image. Then we align the vertices of the model with the pixels of the input image in local regions respectively. Finally, some local shape optimizations are taken to make the model consistent with the input image. Here we define a $N \times M$ matrix A as the coordinates matrix of the model, each row of which represents the three-dimensional coordinates of each vertices.

The normalization is scaling and translating the model to fit the image. We used R feature points which have been detected in the previous section. The distribution of them is shown as Fig.3(c). The R corresponding vertices in the model then can be selected. Let $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_R)^T$ be the matrix of the n feature points of the input image and $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_R)^T$ be the matrix of the corresponding vertices of the face model. $\mathbf{p}_i = (x_i, y_i)^T$ and $\mathbf{v}_i = (x_i', y_i')^T$ represent the X-Y coordinates of the ith feature points in the input image and the model respectively. To center the image as well as the model, we subtract the average value of $\mathbf{P}$ and $\mathbf{V}$ from each $\mathbf{p}_i$ and $\mathbf{v}_i$

$$\mathbf{p_i}' = (x_i - \frac{1}{n}\sum_{k=1}^{n} x_k, y_i - \frac{1}{n}\sum_{k=1}^{n} y_k)^T, \quad (1)$$

$$\mathbf{v_i}' = (x_i' - \frac{1}{n}\sum_{k=1}^{n} x_k', y_i' - \frac{1}{n}\sum_{k=1}^{n} y_k')^T. \quad (2)$$

Then we can get the scaling coefficient s and the offset coef-

ficient **t** as

$$\mathbf{t} = (\frac{1}{n}\sum_{k=1}^{n} x_k - \frac{1}{n}\sum_{k=1}^{n} x_k{}', \frac{1}{n}\sum_{k=1}^{n} y_k - \frac{1}{n}\sum_{k=1}^{n} y_k{}', 0)^T, \quad (3)$$

$$s = \frac{\sum_{i=1}^{n} \|\mathbf{p}_i{}'\|_2}{\sum_{i=1}^{n} \|\mathbf{v}_i{}'\|_2}, \quad (4)$$

where $\|\cdot\|_2$ denotes L2 norm. Let $\mathbf{T} = (\mathbf{t}, \mathbf{t}..., \mathbf{t})^T$ be a matrix whose size is the same as **A**, the coordinates of the model then can be modified as $\mathbf{A}' = s \cdot (\mathbf{A} + \mathbf{T})$. Now the model is holistically aligned with the input image.

The previous works are focus on the holistically adjustment. To obtain more elaborate reconstruction result, local adjustments are taken. We classify *N* vertices of the model into 5 groups including left eye, right eye, nose, mouth and transition points. The operation of local alignment is taken independently for each group. For simplicity, we take the adjustment of mouth region as an example. First, we get the coordinate of the center of mouth in the input image and denote it as $\mathbf{m}_c = (x_c, y_c)^T$. The corresponding vertex in the model is $\mathbf{m}'_c = (x'_c, y'_c)^T$. We calculate the distance between $\mathbf{m}_c$ and $\mathbf{m}'_c$: $\mathbf{r} = \mathbf{m}_c - \mathbf{m}'_c$. Then for each vertex $\mathbf{m}_j = (x_j, y_j)^T$ in the group of mouth, its new coordinate can be calculate as $\mathbf{m}'_j = \mathbf{m}_j + \mathbf{r}$.

Though adjustments have been taken over global and local regions respectively, there still exist some problems because of the individual difference on face components such as the width of nose or mouth. Especially, unsuitable width of the nose in the model will give a greatly bad impact on the finally reconstruction results, which is shown in Fig.3(d)(the highlight regions is in the left corner of the figure). To solve this problem, we perform some local shape optimizations on the region of nose and mouth. Since the operations for both of the mouth and nose regions are similar, we take the process in nose region as an example. In previous sub-section, we have detected several corner points in the nose region. Let $\mathbf{l}_i = (l_{x,i}, l_{y,i})^T$, *i* = 1 or 2, be the coordinates of the feature points on the nose wings. It represents the left wing when *i* = 1 and right wing when *i* = 2. $\mathbf{l}'_i = (l'_{x,i}, l'_{y,i})^T$ are the corresponding vertices in the model. We get

$$\mathbf{d}_{w,i} = \mathbf{l}'_i - \mathbf{l}_i, i = 1, 2 \quad (5)$$

$$\mathbf{d}_{c,i} = \mathbf{l}'_i - \mathbf{n}_c, i = 1, 2 \quad (6)$$

where $\mathbf{n}_c = (x_t, y_t)^T$ represents the tip of nose in the model, $\mathbf{d}_{w,i}$ represents the distance between the wing of nose in the model and the input image, $\mathbf{d}_{c,i}$ represents the distance between the tip of the nose and the wing of nose in the model. Then, for each vertex $\mathbf{n}_j = (x_j, y_j)^T$ in the group of nose, we can have

$$\mathbf{d}_j = \frac{\|\mathbf{n}_j - \mathbf{n}_c\|_2}{\|\mathbf{d}_{c,i}\|_2} \cdot \mathbf{d}_{w,i}, \quad (7)$$

where *i*=1 when $x_j - x_t < 0$, and otherwise, *i*=2. The new coordinate of $\mathbf{n}_j$ can be calculated as $\mathbf{n}'_j = \mathbf{n}_j - \mathbf{d}_j$.

The adjusted model and the final reconstruction result are shown in Fig.3(e) and (f) respectively.

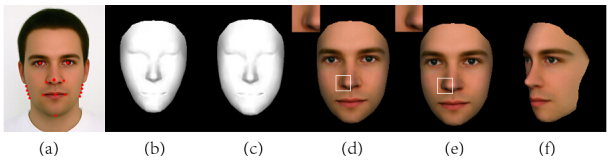


Fig. 3. (a) Feature points used for normalizing, (b) and (c) Two face models, (d) Reconstructed result without local optimization, (e) and (f) Result with local optimization from two angles

2.3. Model processing on Z-axis direction

To make the reconstructed face more desirable, we further adjust the model on Z-axis direction. We present a method of insertion on each surface of the model to obtain a more smooth and dense model. For simplicity, we take the interpolation on the *k*th surface as an example. Since the models we used are both composed of triangular patches, the three vertices of the surface can be denoted as $\mathbf{v}_m = (x_m, y_m, z_m)^T$, *m* = 1, 2, 3. Firstly, we project the surface onto X-Y plane and get a triangle on the plane. Then we interpolate *H* points at regular intervals inside the triangle. For each point $v'_h = (x_h, y_h)$, *h* = 1, 2..., *H*, its Z-coordinate z'_h can be calculated based on the surface identified by $\mathbf{v}_m$.

To make the surface more smooth, we add a smoothing component α_h to z'_h. The distance between $\mathbf{v}'_h$ and the three vertices $\mathbf{v}_m$ can be calculated as $q_m = \|\mathbf{v}'_h - \mathbf{v}_m\|_2$, *m*=1,2,3.

Let D_k be the area of the surface, the value of α_h can be calculated as

$$\alpha_h = \beta \cdot \sqrt{\frac{D_k}{D_{max}}} \cdot q_{min}, \quad (8)$$

where D_{max} represents the area of the largest one of the *M* surface and q_{min} represents the minimum of q_m. β is a empirical coefficient. The modified Z-coordinate of $\mathbf{v}'_h$ can be calculated as $z''_h = z'_h + \alpha_h$.

After inserting points inside all the surfaces, we finally get a dense and smooth model.

3. EXPERIMENTS

In this section, we provide experimental results which demonstrate the effectiveness and efficiency of our method.

3.1. Experimental settings

In Section 2.2, $M = 536$, $N = 548$. The test images in Fig.4(a) were taken by the authors in laboratory with complex background and various illumination. The test images in Fig.4(b) were collected from the Internet with various resolutions and uncertain quality. Before reconstructing the 3D face, all test images are resized to the same width of 400 pixels. In Section 2.2, we set $R = 17$. In Section 2.3, we set $\beta = 15$.

3.2. Results and discussions

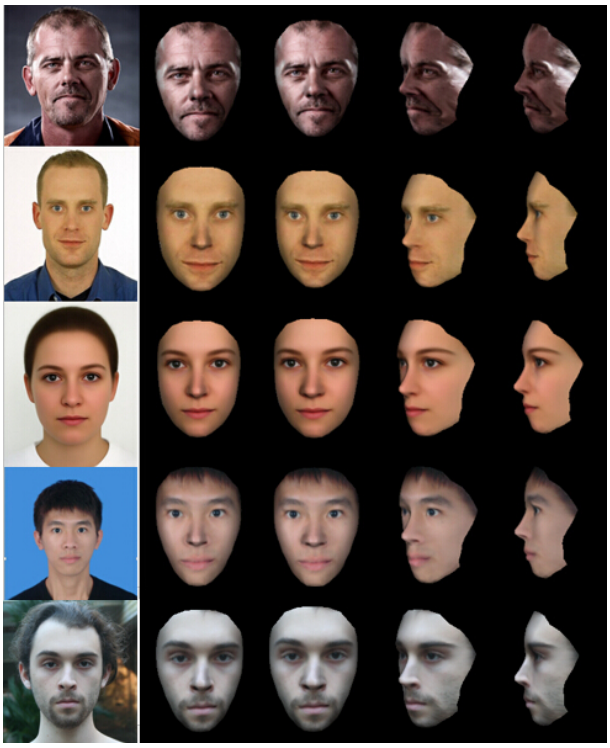
Fig.4 presents the reconstruction results on several test images. The first column in Fig.4 are inputs images. The second column shows the results which do not take local shape optimizations presented in Section 2.2. The other columns present the reconstructed 3D face from three different perspectives. The results shown in Fig.5 demonstrate that our method can obtain good results whether the input images are various resolution or the backgrounds are complex. From the comparison between column 2 and column 3, we can find that the shadow edges in the 3D face are consistent with the ones in the 2D images after taking the local optimizations presented in Section 2.2, which prove the effectiveness of our method. In addition to the effectiveness, efficiency is also important. Since we only use two sparse model and the dense model is obtained in the last step(Section 2.3), most of the calculation processes are simple, which save lots of time. Most learning-based methods cost nearly a hour to match a model with the input from the training examples, e.g.[6]. We test 200 images on a laptop with 2GHz CPU and 4GB memory. The average execution time is 1.5s for reconstructing a 3D face from a single image. To sum up, our method is both effectiveness and efficiency.

4. CONCLUSION

In this paper, we present a fast and robust algorithm for reconstructing 3D face from a single frontal face image. We detect several key points from the input image and then adjust a adjustable 3D face model based on these feature points in X-Y plane and Z-plane respectively. Our algorithm need only a single face image with either complex or simple background as input and the entire process is executed in an automatic manner. It takes only about 1.5 seconds for converting the 2D face image into 3D face. Our approach has been validated experimentally and shows good performance on both robustness and speed.

We attempt to develop a method which can adjust the model outline to fit the face contour. In this case, only one model is needed and the contour of the reconstructed 3D face will be more close to that of face in the input image.

(a)



(b)

Fig. 4. (a) Reconstruction results of images taken by the authors in laboratory (b) Reconstruction results of the images downloaded from the Internet

5. REFERENCES

[1] Yuping Lin, Gérard Medioni, and Jongmoo Choi, "Accurate 3d face reconstruction from weakly calibrated wide baseline images with profile contours," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 1490–1497.

[2] Zhengyou Zhang, Zicheng Liu, Dennis Adler, Michael F Cohen, Erik Hanson, and Ying Shan, "Robust and rapid generation of animated faces from video images: A model-based modeling approach," *International Journal of Computer Vision*, vol. 58, no. 2, pp. 93–119, 2004.

[3] Sung Joo Lee, Kang Ryoung Park, and Jaihie Kim, "A sfm-based 3d face reconstruction method robust to self-occlusion by using a shape conversion matrix," *Pattern Recognition*, vol. 44, no. 7, pp. 1470–1486, 2011.

[4] Joseph J Atick, Paul A Griffin, and A Norman Redlich, "Statistical approach to shape from shading: Reconstruction of three-dimensional face surfaces from single two-dimensional images," *Neural computation*, vol. 8, no. 6, pp. 1321–1340, 1996.

[5] Ira Kemelmacher-Shlizerman and Ronen Basri, "3d face reconstruction from a single image using a single reference face shape," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 394–405, 2011.

[6] Volker Blanz and Thomas Vetter, "A morphable model for the synthesis of 3d faces," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194.

[7] Tal Hassner, "Viewing real-world faces in 3d," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2013, pp. 3607–3614.

[8] Fei Yang, Jue Wang, Eli Shechtman, Lubomir Bourdev, and Dimitri Metaxas, "Expression flow for 3d-aware face component transfer," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4, pp. 60, 2011.

[9] Dalong Jiang, Yuxiao Hu, Shuicheng Yan, Lei Zhang, Hongjiang Zhang, and Wen Gao, "Efficient 3d reconstruction for face recognition," *Pattern Recognition*, vol. 38, no. 6, pp. 787–798, 2005.

[10] Ira Kemelmacher-Shlizerman and Steven M Seitz, "Face reconstruction in the wild," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 1746–1753.

[11] Paul Viola and Michael Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2001, vol. 1, pp. I–511.

[12] Paul Viola and Michael J Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[13] Jongmoo Choi, Gérard Medioni, Yuping Lin, Luciano Silva, Olga Regina, Mauricio Pamplona, and Timothy C Faltemier, "3d face reconstruction using a single or multiple views," in *International Conference on Pattern Recognition (ICPR)*. IEEE, 2010, pp. 3959–3962.

[14] Chris Harris and Mike Stephens, "A combined corner and edge detector.," in *Alvey vision conference*. Citeseer, 1988, vol. 15, p. 50.

[15] Rein-Lien Hsu, Mohamed Abdel-Mottaleb, and Anil K Jain, "Face detection in color images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 696–706, 2002.