Reconstruction of Facial Shape from Freehand Multi-viewpoint Snapshots

Seiji Suzuki[1], Hideo Saito[1], and Masaaki Mochimaru[2]

[1] Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa, Japan
{suzuki,saito}@hvrl.ics.keio.ac.jp
[2] Advanced Industrial Science and Technology, 2-41-6 Aomi, Koto-ku, Tokyo, Japan
m-mochimaru@aist.go.jp

Abstract. We propose a method that can reconstruct both a 3D facial shape and camera poses from freehand multi-viewpoint snapshots. This method is based on Active Shape Model (ASM) using a facial shape database. Most ASM methods require an image in which the camera pose is known, but our method does not require this information. First, we choose an initial shape by selecting the model from the database which is most suitable to input images. Then, we improve the model by morphing it to fit the input images. Next, we estimate the camera poses using the morphed model. Finally we repeat the process, improving both the facial shape and the camera poses until the error between the input images and the computed result is minimized. Through experimentation, we show that our method reconstructs the facial shape within 3.5 mm of the ground truth.

1 Introduction

3D shape reconstruction is one of the research issues that is extensively studied for over 20 years in computer vision. Hardware devices such as a range scanner help us to measure a 3D shape accurately [1]. A video projector, one of the hardware devices, can also help us to reconstruct a 3D shape by projecting some particular patterns to an object [2,3]. However, these hardware devices are expensive and not easy to use. Therefore, a lot of image based techniques which do not require any hardware devices except camera have been presented.

For example, *Shape from Shading* and *Shape from Texture* can reconstruct a 3D shape from a single-viewpoint image. These methods, however, have so many constraints on reflectance properties and illumination conditions that it could not be used easily. *Stereo Vision* which requires multi-viewpoint images is also hard to use because the user have to calibrate cameras accurately.

There is a technique named *Structure from Motion* [4] that reconstructs a 3D shape from sequential images such as video. This technique uses *Optical Flow* in order to find corresponding points between subsequent frames. However, *Optical Flow* can not be computed accurately because cheeks have uniform texture.

To solve these problems, methods based on *Active Shape Model* (ASM) have developed. These methods uses a database which has 3D facial shapes measured

G. Bebis et al. (Eds.): ISVC 2009, Part II, LNCS 5876, pp. 641–650, 2009.

by a range scanner. A 3D facial shape can be reconstructed accurately using Principal Component Analysis (PCA) of the database. Some traditional methods [5, 6, 7], however, use only one input image, so 3D geometric information is ignored. Even though the appearance of their results is plausible, the reconstructed shape may lack geometric compliance.

In contrast, the other ASM methods [8, 9, 10] consider the geometric information, using multi-viewpoint images. However, camera poses are assumed to be already known. This assumption requires the user to calibrate cameras. Recently, a method in which the user need not calibrate any cameras was proposed [11]. A 3D facial shape can be reconstructed from uncalibrated multi-viewpoint images. However, this method reconstructs a shape without estimating the camera poses. Therefore the reconstructed results is not accurate.

In this paper, we propose a method that can reconstruct a 3D facial shape from uncalibrated multi-viewpoint snapshots. In fact, our method requires some manual inputs such as clicking facial feature points. However, the user do not have to prepare any special hardware devices and any calibrations, so this method is easy to use even at home.

2 Method

We aim to recover a facial shape from uncalibrated multi-viewpoint snapshots, which capture a face from various unknown poses. This means that we need to estimate both a facial shape and camera poses from the input images. However, the shape optimization requires camera poses, while the pose estimation requires a facial shape. Even though an initial shape is quite different from a real shape, it is only one information that can be used for the pose estimation. That is why the poses and the shape can not be accurately estimated simultaneously in one computation. Therefore, in the proposed method, we designed an iterative algorithm for reducing estimation error.

Fig. 1 shows a flowchart of this method. We roughly estimate a camera pose against a target human face in each input image using an interim shape. A projected image of the interim shape can be rendered at the each estimated pose. The error between the input images and the projected images is computed by error functions. The interim shape is optimized to minimize the error.

The interim shape is quite different from a real target shape at the beginning of this process. As the interim shape is fitted to the real shape, the poses can be computed more accurately. The accurate poses make the interim shape more accurate. To repeat this process, we can get a reconstructed shape as the optimized interim shape.

Our method can be regarded as an energy minimization. We want to find the shape $\boldsymbol{x}$ which minimizes the error y in the equation $y = f(\boldsymbol{x})$, where f is the error function. It is better that the argument vector $\boldsymbol{x}$ is lower dimensional in this situation. We use PCA of a facial shape database to make $\boldsymbol{x}$ lower dimensional vector.

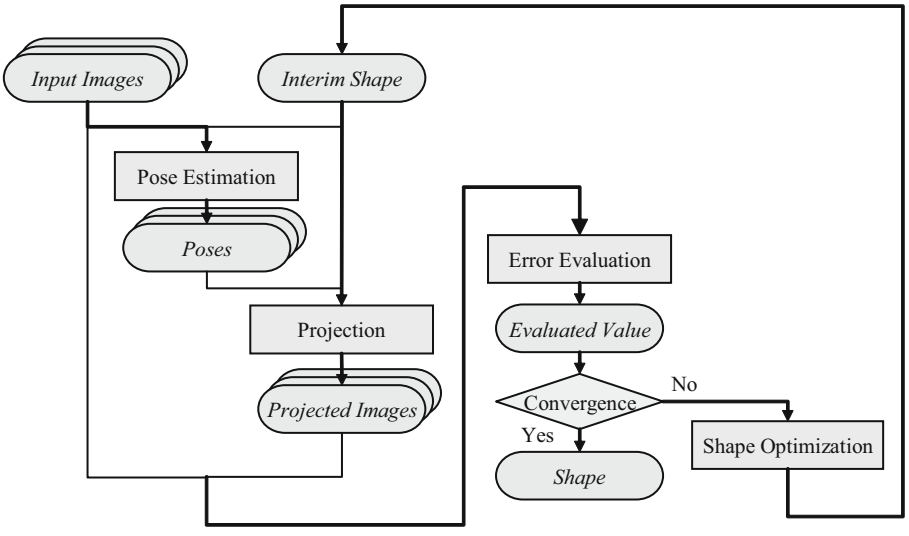


Fig. 1. Flowchart of this method

We describe about the database in 2.1, and PCA in 2.2. The way of the initial shape generation, the camera pose estimation and the facial shape optimization are referred in 2.3, 2.4, 2.5 respectively. The definition of the error functions is in 2.6.

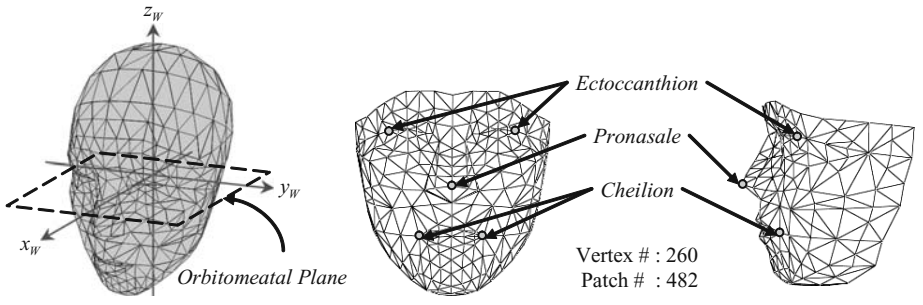
2.1 Database

Our database is composed of human's head shape data. Each data is scanned by a range scanner. The scanned data have around 200 thousand vertices. An anatomist extracted one hundred vertices which have an anatomically important information, and another 330 vertices are interpolated. Fig. 2(a) shows the orbitomeatal plane coordinate system, on which all the human's head shapes are defined. We use only a facial part of the 430 head vertices. The facial part has 260 vertices shown in Fig. 2(b). We have two databases which include 52 males' and 52 females' facial shape respectively.

2.2 Principal Component Analysis

The facial shape, that is represented as a multidimensional vector, should be lower dimensional through the optimization. PCA can compress the multidimensional vector to lower dimensional one.

Our database has m persons' shape vectors. Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m$ denote them respectively. Each vector is defined as $\boldsymbol{x} = [x_1, y_1, z_1, x_2, y_2, z_2, \cdots, x_n, y_n, z_n]^\top$, where n is the number of vertices. In our database, $n = 260$ and $m = 52$.



(a) Orbitomeatal plane (b) Facial part model

Fig. 2. Database definition

PCA calculates eigenvectors $\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_k$ from the shape vectors, where $k = \min(3n, m)$ and $\boldsymbol{p}_i \in \Re^{3n}$ $(1 \leq i \leq k)$. An arbitrary facial shape $\boldsymbol{x}$ can be represented by the eigenvectors and an average shape $\bar{\boldsymbol{x}} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}_i$:

$$\boldsymbol{x} = \bar{\boldsymbol{x}} + \sum_{i=1}^{k} s_i \boldsymbol{p}_i. \tag{1}$$

The vector $\boldsymbol{s} = [s_1, s_2, \cdots, s_k]^\top$ is in a one-to-one correspondence with the vector $\boldsymbol{x}$. Both $\boldsymbol{x}$ and $\boldsymbol{s}$ represent the facial shape uniquely.

Choosing l $(1 \leq l < k)$ elements from $\boldsymbol{s}$ in ascending order, we get a vector $\hat{\boldsymbol{s}} = [s_1, s_2, \cdots, s_l]^\top$. If the shape is represented by $\hat{\boldsymbol{s}}$, a predicted shape $\hat{\boldsymbol{x}}$ can be computed:

$$\hat{\boldsymbol{x}} = \bar{\boldsymbol{x}} + \sum_{i=1}^{l} s_i \boldsymbol{p}_i. \tag{2}$$

Thus we can convert $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{s}}$ each other at any time.

The facial shape is represented by the $3n$ dimensional vector $\boldsymbol{x}$. Now it is compressed to the l dimensional vector $\hat{\boldsymbol{s}}$ by the theory of PCA, so it can be practical to optimize the facial shape.

2.3 Initial Shape

We have to choose an initial shape from our database. To choose the most suitable shape, first we project all shapes $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m$ to the eigenspace. Next we estimate a camera pose using the projected shape $\hat{\boldsymbol{s}}_i$ $(1 \leq i \leq m)$ in each input image, and then render the projected image of each shape. Finally we compute the evaluated values $y_i = f(\hat{\boldsymbol{s}}_i)$ and determine an initial shape $\hat{\boldsymbol{s}}_{\text{init}}$ as the shape $\hat{\boldsymbol{s}}_i$ which has the minimum error y_i.

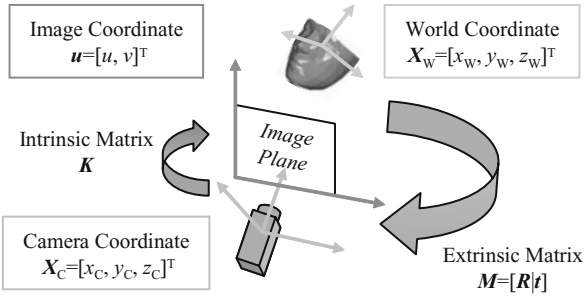


Fig. 3. Relationships among coordinate systems

2.4 Pose Estimation

Fig. 3 shows image coordinate system $\boldsymbol{u} = [u, v]^\top$, camera coordinate system $\boldsymbol{X}_C = [x_C, y_C, z_C]^\top$, and world coordinate system $\boldsymbol{X}_W = [x_W, y_W, z_W]^\top$. They relate to each other:

$$\tilde{\boldsymbol{u}} \simeq \boldsymbol{K}\tilde{\boldsymbol{X}}_C \tag{3}$$

$$\tilde{\boldsymbol{X}}_C \simeq \boldsymbol{M}\tilde{\boldsymbol{X}}_W, \tag{4}$$

where $\boldsymbol{K}$ is intrinsic camera parameter and $\boldsymbol{M} = [\boldsymbol{R}|\boldsymbol{t}]$ is extrinsic camera parameter. They are denoted vy following elements:

$$\boldsymbol{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \ \boldsymbol{M} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}. \tag{5}$$

The world coordinate system equals to the orbitomeatal plane coordinate system in this situation. We assume that the intrinsic parameter $\boldsymbol{K}$ is already known, and the 2D feature points $\boldsymbol{u}$ are also known because of user's clicking. The 3D facial feature points $\boldsymbol{X}_W$ is given by the interim shape $\hat{\boldsymbol{s}}$.

In this method we compute the camera pose, that is the extrinsic parameter $\boldsymbol{M}$, from a pair of the five feature points $\boldsymbol{u}$ and $\boldsymbol{X}_W$ (see Fig. 2) using Zhang's method [12].

2.5 Shape Optimization

Eq. (3) (4) leads to a following equation:

$$\begin{aligned} \tilde{\boldsymbol{u}} &\simeq \boldsymbol{K}\boldsymbol{M}\tilde{\boldsymbol{X}}_W \\ &\simeq \boldsymbol{P}\tilde{\boldsymbol{X}}_W, \end{aligned} \tag{6}$$

where $\boldsymbol{P}$ is so-called projection matrix. We can project the interim shape $\hat{\boldsymbol{s}}$ to an image at the same pose as the input camera pose $\boldsymbol{P}$, so the projected

image depends on the projection matrix $\boldsymbol{P}$ and the interim shape $\hat{\boldsymbol{s}}$. In fact, the projection matrix is computed from the interim shape. Therefore, as a result, the projected image relies on only the interim shape.

We can compute the error between the input image and the projected image. The shape is otimized to minimize it. We use *Levenberg-Marquardt* method for the optimization. This algorithm generates the optimized shape $\hat{\boldsymbol{s}}_{\text{opt}} = \arg\min_{\hat{\boldsymbol{s}}} f(\hat{\boldsymbol{s}})$, where f denotes the error function. In fact, the error function is composed of the four functions described at 2.6:

$$f(\hat{\boldsymbol{s}}) = \sum_{i=1}^{4} \{\alpha_i f_i(\hat{\boldsymbol{s}})\}^2, \tag{7}$$

where α_i $(1 \leq i \leq 4)$ is a weight coefficient of each function determined empirically. The optimized shape $\hat{\boldsymbol{s}}_{\text{opt}}$ is the final output of this method.

2.6 Error Functions

There are four error functions. They require manual inputs such as five feature points $\boldsymbol{u}_{\text{input}}$, a silhouette $\boldsymbol{S}_{\text{input}}$, and an outline $\boldsymbol{L}_{\text{input}}$ on each input image $\boldsymbol{I}_{\text{input}}$. The silhouette is a facial part of the input image. The outline is a part of the contour of the silhouette. A border between a face and a background is the outline. In contrast, a border between a face and hair is not the outline.

Facial Likelihood Error Function. This function computes a correlation value between an argument shape $\hat{\boldsymbol{s}}$ and the set of the database shapes:

$$f_1(\hat{\boldsymbol{s}}) = 1 - \exp\left(-\frac{1}{2}\sum_{i=1}^{l}\frac{s_i^2}{\lambda_i}\right), \tag{8}$$

where λ_i $(1 \leq i \leq l)$ denotes the eigenvalue, that is a variance of the learning data set. We suppose that the human facial shapes are on the Gaussian distribution from the average shape.

If the argument shape $\hat{\boldsymbol{s}}$ is far from the data set, the error value should be high. This function prevent the interim shape from being morphed too much and being far from humanity.

Facial Contour Error Function. This function evaluates the difference in terms of the contour between the argument shape and the input image: The definition is:

$$f_2(\hat{\boldsymbol{s}}) = \frac{\iint_A d_{\text{proj}}^{(u,v)}\, dudv}{w \iint_A dudv}, \tag{9}$$

where

$$A = \left\{(u,v) \,|\boldsymbol{L}_{\text{input}}^{(u,v)} = \text{white}\right\} \tag{10}$$

$$d_{\mathrm{proj}}^{(u,v)} = D\left(\boldsymbol{S}_{\mathrm{proj}}\right)^{(u,v)} + D\left(\tilde{\boldsymbol{S}}_{\mathrm{proj}}\right)^{(u,v)} \tag{11}$$

$$w = \frac{\sqrt{f_x^2 + f_y^2}}{t_3}. \tag{12}$$

$\tilde{\boldsymbol{S}}$ denotes a negative image of a binary image $\boldsymbol{S}$. $D(\boldsymbol{S})$ means the euclidean distance transformation of $\boldsymbol{S}$. Note that the silhouette $\boldsymbol{S}_{\mathrm{proj}}$ is the projection of the argument shape $\hat{\boldsymbol{s}}$, so it depends on $\hat{\boldsymbol{s}}$. The main part of this function is $\iint_A d_{\mathrm{proj}}^{(u,v)} dudv$. This part means the sum of absolute distance from the outline to the projected silhouette. The denominator is a normalization factor, where w represents the projected facial area size and $\iint_A dudv$ is the length of $\boldsymbol{L}_{\mathrm{input}}$.

Feature Points Error Function. This function evaluates a 2D distance of the feature points between the input image and the projected image. Let $\boldsymbol{u}_{\mathrm{input}}, \boldsymbol{u}_{\mathrm{proj}}$ denote the feature points on each image:

$$f_3\left(\hat{\boldsymbol{s}}\right) = \frac{1}{5w} \sum_{i=1}^{5} \left\| \boldsymbol{u}_{\mathrm{input}}^{(i)} - \boldsymbol{u}_{\mathrm{proj}}^{(i)} \right\|, \tag{13}$$

where $\boldsymbol{u}^{(i)}$ $(1 \leq i \leq 5)$ represents the coordinate of each feature point. Note that the projected points $\boldsymbol{u}_{\mathrm{proj}}$ depends on $\hat{\boldsymbol{s}}$. The denominator is a normalization factor, where w is defined by Eq. (12).

Texture Error Function. This function can evaluate the detail of the face. We use the most frontal facial input image for a texture, and then render a texture mapped projected image $\boldsymbol{I}_{\mathrm{proj}}$ onto the other input image $\boldsymbol{I}_{\mathrm{input}}$. We compute the error usihng following equation:

$$f_4\left(\hat{\boldsymbol{s}}\right) = \frac{\iint_B \left\| \boldsymbol{I}_{\mathrm{input}}^{(u,v)} - \boldsymbol{I}_{\mathrm{proj}}^{(u,v)} \right\| dudv}{\iint_B dudv}, \tag{14}$$

where

$$B = \left\{ (u,v) \,|\, \boldsymbol{S}_{\mathrm{input}}^{(u,v)} = \boldsymbol{T}_{\mathrm{proj}}^{(u,v)} = \mathrm{white} \right\}. \tag{15}$$

$\boldsymbol{T}_{\mathrm{proj}}$ is a mask image that presents a texture mapped area.

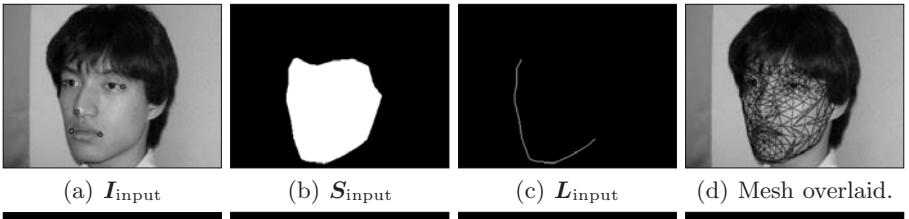
The numerator is the sum of absolute difference of each appearance in a region of comparable area. The denominator is a normalization term, that is a size of the comparable area.

3 Experiments

First, We reconstruct a facial shape from multi-viewpoint snapshots. The result is show in 3.1. Next, in order to discuss the accuracy, we reconstruct five persons' shapes whose real shapes are measured by a range scaner in advance. The accuracy is shown in 3.2 and discussed in 3.3.

3.1 Reconstruction

The number of input images is four in 640×480 resolution. The input images are taken by a digital still camera and not calibrated extrinsically. Through the experimentation, we use $l = 20$ principal components, which enable the cumulative contribution ratio 90%. The parameters in Eq. (7) is determined empirically as $\alpha_1 = 1.0$, $\alpha_2 = 6.0$, $\alpha_3 = 3.0$, $\alpha_4 = 0.1$. We decide $\alpha_i \, (1 \leq i \leq 4)$, where the evaluated values $\alpha_i f_i \, (\hat{\boldsymbol{s}})$ are nearly same to each other.



(a) $\boldsymbol{I}_{\text{input}}$ (b) $\boldsymbol{S}_{\text{input}}$ (c) $\boldsymbol{L}_{\text{input}}$ (d) Mesh overlaid.

(e) Proj. (f) $\boldsymbol{S}_{\text{proj}}$ (g) $\boldsymbol{I}_{\text{proj}}$ (h) $\boldsymbol{T}_{\text{proj}}$

Fig. 4. Input image and reconstructed result

Fig. 4 shows the results. Though the number of the input images is four, we show one typical image in the figure. Fig. 4(a) is the input image with cliked feature points. Fig. 4(b) and Fig. 4(c) are the silhouette and the outline of the input image respectively. We reconstruct a shape from these inputs. The result is shown in Fig. 4(e). Fig. 4(f) shows the silhouette image of it. Fig. 4(g) is the projected image with a texture, and Fig. 4(h) is the mask image that means the texture mapped area. Fig. 4(d) shows an image with a reconstructed mesh.

To compare Fig. 4(a) with Fig. 4(g), the PSNR (Peak Signal-to-Noise Ratio) of the appearance is 24.4 dB. The computation time is around 10 minutes in the condition of Windows XP SP3, Intel Core 2 Duo 6700 (2.66GHz), 3.5GB RAM.

3.2 Comparison with Range Data

We reconstruct five persons' shapes whose real shapes are already known by a range scanner. They consist of three males and two females, and we use a respective database. The experimental condition is the same as 3.1.

We computed 3D reconstruction errors from real shapes. The error means the average of the euclidean distance between the true verteces and their

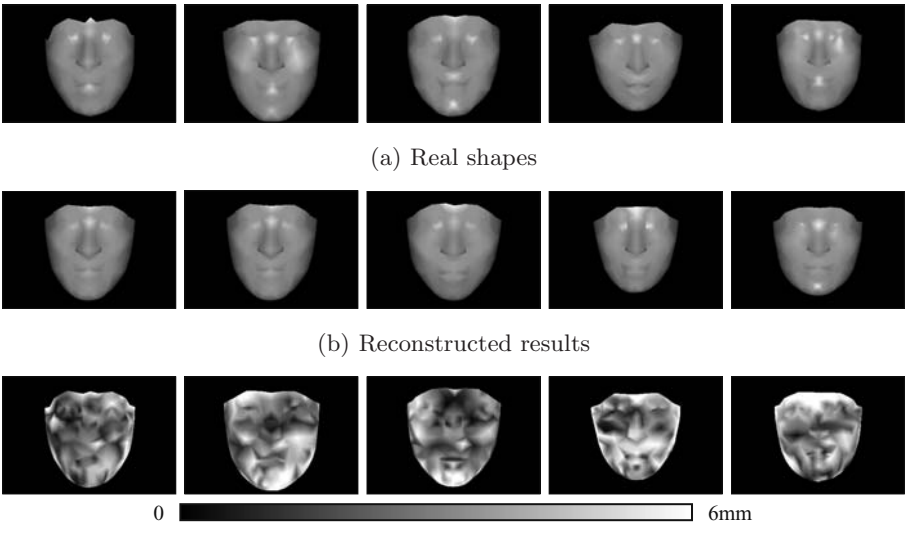
Table 1. Shape evaluation. Each column corresponds with the each person. The top row shows the error between the real shape and the reconstructed shape. The middle row means the error between the real shape and the most similar shape in the database. The bottom row is the average value of errors between the real shape and the respective shapes in the database (mm).

Person ID	Male 1	Male 2	Male 3	Female 1	Female 2
Reconstructed	3.1	3.3	3.2	3.9	3.9
Min DB	3.5	3.2	2.9	3.0	2.7
Avg DB	4.9	5.0	4.3	4.6	4.7

corresponding reconstructed vertex positions. Table 1 shows the fact that the reconstructed shape is around 3.5 mm different from the real shape. Compared with the middle row and the bottom row, the reconstructed results look suitable.

3.3 Error Distribution

Fig. 5 shows reconstruction error maps. Each column corresponds with the each person. Fig. 5(a) is the real shapes which is measured by a range scanner. Fig. 5(b) is the reconstructed shape, and Fig. 5(c) is the error maps.



(a) Real shapes

(b) Reconstructed results

(c) Error maps

Fig. 5. Reconstructed results and error maps

The center of the face has a small error. In contrast, the edge has a big error. This occurs because the feature points are concentrated to the center of the face. That is why the edge part can not be computed accurately.

4 Conclusion

We proposed a method that can reconstruct both a facial shape and camera poses from freehand multi-viewpoint snapshots. This method does not use any special hardware device. The most of conventional methods require a calibrated multi camera system, but our method does not require it because we estimate both of them simultaneously.

The reconstruction error is around 3.5 mm. However, our method needs manual inputs such as facial feature points, a silhouette and an outline. It is better to decrease these manual inputs. It will be a future task.

References

1. Brunsman, M.A., Daanen, H.A.M., Robinette, K.M.: Optimal postures and positioning for human body scanning. In: Proc. of Int'l Conf. on Recent Advances in 3-D Digital Imaging and Modeling, pp. 266–273 (1997)
2. Zhang, L., Snavely, N., Curless, B., Seitz, S.M.: Spacetime faces: High resolution capture for modeling and animation. ACM Trans. on Graphics 23, 548–558 (2004)
3. Siebert, J.P., Marshall, S.J.: Human body 3d imaging by speckle texture projection photogrammetry. Sensor Review 20, 218–226 (2000)
4. Chowdhury, A.K.R., Chellappa, R.: Face reconstruction from monocular video using uncertainty analysis and a generic model. Computer Vision and Image Understanding 91, 188–213 (2003)
5. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: Proc. of the 26th Annual Conf. on Computer Graphics and Interactive Techniques, pp. 187–194 (1999)
6. Romdhani, S., Blanz, V., Vetter, T.: Face identification by fitting a 3d morphable model using linear shape and texture error functions. In: Proc. of the Seventh European Conf. on Computer Vision, vol. 4, pp. 3–19 (2002)
7. Blanz, V., Vetter, T.: Face recognition based on fitting a 3d morphable model. IEEE Trans. on Pattern Analysis and Machine Intelligence 25 (2003)
8. Hu, Y., Jiang, D., Yan, S., Zhang, L., Zhang, H.: Automatic 3d reconstruction for face recognition. In: Proc. of the Sixth IEEE Int'l Conf. on Automatic Face and Gesture Recognition, pp. 843–848 (2004)
9. Jiang, D., Hu, Y., Yan, S., Zhang, L., Zhang, H., Gao, W.: Efficient 3d reconstruction for face recognition. Pattern Recognition 38 (2005)
10. Amberg, B., Blake, A., Fitzgibbon, A., Romdhani, S., Vetter, T.: Reconstructing high quality face-surfaces using model based stereo. In: Proc. of the Eleventh IEEE Int'l Conf. on Computer Vision (2007)
11. Takeuchi, T., Saito, H., Mochimaru, M.: 3d-face model reconstruction utilizing facial shape database from multiple uncalibrated cameras. In: Proc. of the 16th Int'l Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision (2008)
12. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. on Pattern Analysis and Machine Intelligence 22, 1330–1334 (2000)