

Article

3D Face Modeling Using the Multi-Deformable Method

Jinkyu Hwang, Sunjin Yu, Joongrock Kim and Sangyoun Lee *

Department of Electrical and Electronics Engineering, Yonsei University, 134 Shinchon-dong, Seodaemun-gu, Seoul 120-749, Korea; E-Mails: winispirit@yonsei.ac.kr (J.H.); biometrics@yonsei.ac.kr (S.Y.); jurock@yonsei.ac.kr (J.K.)

* Author to whom correspondence should be addressed; E-Mail: syleee@yonsei.ac.kr; Tel.: +82-2-2123-5768.

Received: 24 July 2012; in revised form: 14 September 2012 / Accepted: 19 September 2012 / Published: 25 September 2012

Abstract: In this paper, we focus on the problem of the accuracy performance of 3D face modeling techniques using corresponding features in multiple views, which is quite sensitive to feature extraction errors. To solve the problem, we adopt a statistical model-based 3D face modeling approach in a mirror system consisting of two mirrors and a camera. The overall procedure of our 3D facial modeling method has two primary steps: 3D facial shape estimation using a multiple 3D face deformable model and texture mapping using seamless cloning that is a type of gradient-domain blending. To evaluate our method's performance, we generate 3D faces of 30 individuals and then carry out two tests: accuracy test and robustness test. Our method shows not only highly accurate 3D face shape results when compared with the ground truth, but also robustness to feature extraction errors. Moreover, 3D face rendering results intuitively show that our method is more robust to feature extraction errors than other 3D face modeling methods. An additional contribution of our method is that a wide range of face textures can be acquired by the mirror system. By using this texture map, we generate realistic 3D face for individuals at the end of the paper.

Keywords: 3D face modeling; face deformable model; face shape estimation; statistical face model

1. Introduction

Three-dimensional (3D) face modeling is a challenging topic in computer graphics and computer vision. Unlike 2D face models, 3D face models can realistically express face deformation and pose variation with depth information. With these advantages, 3D face models have been applied to various applications, including movies, 3D animation and telecommunications [1,2].

Three dimensional modeling systems can be categorized into active and passive vision systems [3]. An active vision system calculates 3D information by measuring a beam of light radiated from an external device such as a beam projector or laser. The typical 3D face modeling method using an active vision system constructs a 3D face mesh using a captured 3D point cloud [4–9]. In such systems, a 3D laser scanner or calibrated stereo camera with structured light can be used to capture 3D coordinates and texture information. While these methods are highly accurate, they are also time consuming, and the necessary equipment is expensive.

Nowadays, the passive vision-based 3D modeling system is preferred for human faces because the glare from light-emitting devices can be unpleasant for the users. Passive vision-based system means a system that needs no light-emitting devices and estimates 3D information from 2D images. In passive vision-based 3D face modeling, 3D information can be calculated by analyzing camera geometry from corresponding features in multiple views [10] or adjusting the statistical 3D face model to captured facial images [2,11]. For convenience, we call the former the corresponding feature-based 3D face modeling method and the latter the statistical model-based 3D face modeling method.

Among the 3D face modeling methods using the passive vision system, the most commonly used one is the corresponding feature-based 3D face modeling method. This method is less computationally expensive because it uses only a few feature points to generate a 3D facial shape. Additionally, this method can generate highly accurate 3D facial shapes by using real 3D information calculated from the camera geometry. However, the accuracy of the 3D facial shapes declines rapidly if the extracted locations of the corresponding points are not exact. This problem should be solved to apply to automatic 3D modeling system because even excellent feature extraction techniques such as the active appearance model (AAM) [12] and the active shape model (ASM) [13,14] can produce erroneous feature extraction results for indistinct parts of the face.

In this paper, we aim to develop a realistic 3D face modeling method that is robust to feature extraction errors and generates accurate 3D face modeling results. To achieve this, we propose a novel 3D face modeling method which has two primary steps: 3D facial shape estimation and texture mapping with a texture blending method.

In the 3D facial shape estimation procedure, we take a statistical model-based 3D face modeling approach as a fundamental concept. Among the statistical model-based methods, we use in particular the deformable face model that utilizes location information of facial features in the input image. This method is robust to feature extraction errors because it uses pre-trained 3D face data to estimate 3D facial shapes from input face images but it is a little less accurate than the corresponding feature-based 3D face modeling methods. To improve accuracy of the 3D facial shapes, we propose a 3D face shape estimation method using multiple 3D face deformable models.

In the texture mapping procedure, we apply a cylindrical mapping and a stitching technique to generate a texture map. When stitching each face part, we apply a modified gradient-domain blending

technique [15] to remove the seam that appears at the boundaries of each face part because of photometric inconsistency.

This paper is organized as follows: in Section 2, we introduce previous 3D face modeling techniques and the 3D face deformable model that is basis of proposed method. In Section 3, we address our 3D facial shape estimation method with mirror system. In Section 4, we describe our texture mapping and texture map generation method using a modified gradient-domain blending technique. Then, we discuss the 3D face modeling results and evaluate our method's performances compared with those of other 3D face modeling methods and ground truth in Section 5. Finally, we conclude our paper and address future work in Section 6.

2. Preliminary Study

In this section, we address previous works and the fundamental concept of the proposed 3D face modeling method. In Sections 2.1 and 2.2, we introduce previous 3D face modeling methods using passive vision systems. We categorize them into two groups: corresponding feature-based 3D face modeling and statistical model-based 3D face modeling. Then, we study strengths and weaknesses of these methods. In Section 2.3, we concretely describe the 3D face modeling method using a deformable model which is a type of statistical model-based 3D face modeling because it is a fundamental concept of the proposed method that will be described in Section 3.1.

2.1. Corresponding Feature-Based 3D Face Modeling

The simplest and fastest way to generate a 3D face model using the corresponding features is to use orthogonal views [16–18]. In this method, the 3D coordinates of the features can be easily calculated from manually selected feature points in two orthogonal views of the face. This method is quite easy to implement, but orthogonality between the two views is necessary.

Some researchers construct 3D faces from several facial images. Fua *et al.* [19] proposed a regularized bundle-adjustment on triplet images to reconstruct 3D face models from image sequences. Their method takes advantage of a rough knowledge of the head's shape (from a generic face model), but it is computationally expensive because it requires dense stereo matching. In a similar approach, Lee *et al.* [20] constructed a 3D head model using two-pass bundle adjustments. In the first pass, the method computes several feature points of a target 3D head and then uses these features to obtain a roughly matched head model by modifying a generic head. Next, the second pass bundle adjustment is carried out to obtain a detailed 3D head model. Pighin *et al.* [21] developed a method that generates 3D faces by fitting a generic 3D face model on pre-defined 3D landmark points which are reconstructed from sequentially captured facial images. The pre-defined 3D landmark points can be calculated by the structured from motion (SfM) method after extracting the corresponding feature points in images captured from different views. The texture map is generated by combining the multi-view photographs into a cylindrical map. In their method, face images from different views provide a wide range of textures and features (e.g., the ears) which can be used to generate a realistic 3D face. However, the method is somewhat inconvenient because the user must hold a stationary pose during image capture. In addition, the 3D reconstruction is quite sensitive to the accuracy of the feature extraction so that a manually intensive procedure is required.

As another example of a corresponding feature-based method, Lin *et al.* [22] proposed a 3D face modeling system using two mirrors positioned next to the face to simultaneously capture three different views of the face. Then, they reconstruct 3D points annotated with markers by analyzing the relation between directly captured markers and the markers reflected in the mirrors. In their method, a wide range of face data can be acquired from the captured face images. In addition, this approach avoids the problem of synchronization of multiple cameras, although it is still sensitive to feature extraction error.

2.2. Statistical Model-Based 3D Face Modeling

In 3D face modeling using statistical model, the 3D morphable face model suggested by Blanz and Vetter [23] is the most well-known. To generate a 3D morphable face model, they construct a database including the 3D coordinates and skin texture from a real human face captured by a 3D laser scanner. Then, statistical analysis is carried out to determine control parameters for the 3D face shape and skin texture deformation. During the modeling procedure, the model parameters are iteratively adjusted in order to fit the model to the input image. This gives remarkably realistic results, but the computational cost is very high.

To improve the speed, researchers have proposed 3D face modeling methods using a single-view image. Kuo *et al.* [24,25] proposed a method to synthesize a lateral face from a single frontal-view image. They construct a facial image database containing both frontal and lateral views and then define anthropometric parameters that represent the distance between two features manually extracted by anthropometric definition. In the modeling stage, they estimate the lateral facial parameters from the input frontal image using the relationship between the frontal and lateral facial parameters. Baek *et al.* [26] suggested an anthropometry-based 3D face modeling technique. They built a database after measuring anthropometric information from anatomically meaningful 3D points among a 3D point cloud captured by a 3D laser scanner. Then, they created a statistical model to control the overall 3D face shape after statistical analysis of the database. This method is much faster than the 3D morphable face model, but the depth estimation results are sensitive to head poses, which may result in inaccurate distance measurement between landmarks. Importantly, this approach is also limited to the reconstruction of frontal views.

2.3. 3D Face Shape Model Generation

The 3D face deformable model is a type of parametric model that can deform shapes and textures by changing related parameters. The morphable face model [27], which generates a textured 3D face by controlling parameters that can be acquired from statistical analysis of 3D face scans containing geometric and textural data, is the most representative model in this class.

Generally, the computational costs of morphable face models are very high because they use entire face data (vertices and texture) and require many parameters to fit on input face images. On the other hand, a 3D face deformable model is less computationally expensive because it uses only geometric information (*i.e.*, the 3D coordinates of 3D face scans) and is composed of sparsely distributed vertices rather than full vertices of 3D scans. For clarity, the 3D deformable model that is composed of sparsely distributed vertices is called the 3D face shape model (FSM) in this paper. The 3D FSM is composed

of anatomically meaningful 3D vertices which can represent the shape of the entire face. Each vertex is called a 3D FSM landmark, and a set of landmarks is called a 3D FSM face shape.

The deformation of 3D FSM can be carried out by global and local deformations. In a global deformation, the position and shape of the 3D FSM can be determined by a 3D affine transformation. The 3D affine transformation can be expressed as a 3×1 translation matrix (T) and a 3×3 matrix (A) of rotation and skew transformations. Under a global deformation, the coordinates of a vertex ($[X, Y, Z]$) can be transformed into ($[X_t, Y_t, Z_t]$) by:

$$\begin{bmatrix} X_t \\ Y_t \\ Z_t \end{bmatrix} = \begin{bmatrix} A_1 & A_2 & A_3 \\ A_4 & A_5 & A_6 \\ A_7 & A_8 & A_9 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} \quad (1)$$

In a local deformation, a detailed deformation of 3D FSM can be parameterized by statistical analysis. First, 3D face shape data sets are constructed from 3D vertices corresponding to the position of 3D FSM landmarks in each 3D face scan. These sets (S) can be represented as:

$$\begin{aligned} S_i &= [x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_N, y_N, z_N]^T \\ S &= [S_1, S_2, \dots, S_M]^T \end{aligned} \quad (2)$$

where M is the total number of face shape sets and N is the number of landmarks of 3D FSM. After constructing the 3D face shape data sets, a shape alignment procedure is carried out using 3D procrustes analysis [28], and the mean face shape is calculated as:

$$\bar{S} = \frac{1}{M} \sum_{i=1}^M S_i \quad (3)$$

Then, feature vectors of the face data sets are calculated by principal component analysis (PCA). In PCA, the feature vectors of the shape data sets are the eigenvectors (Φ) of the covariance matrix (Σ) of the normalized shape data set (D):

$$\begin{aligned} \Gamma_i &= S_i - \bar{S} \\ \Sigma &= D D^T, D = [\Gamma_1, \Gamma_2, \dots, \Gamma_M] \\ \Sigma \Phi &= \Phi \Lambda \end{aligned} \quad (4)$$

where Λ denotes a diagonal matrix of eigenvalues:

$$\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_{3M} \end{bmatrix} \quad (5)$$

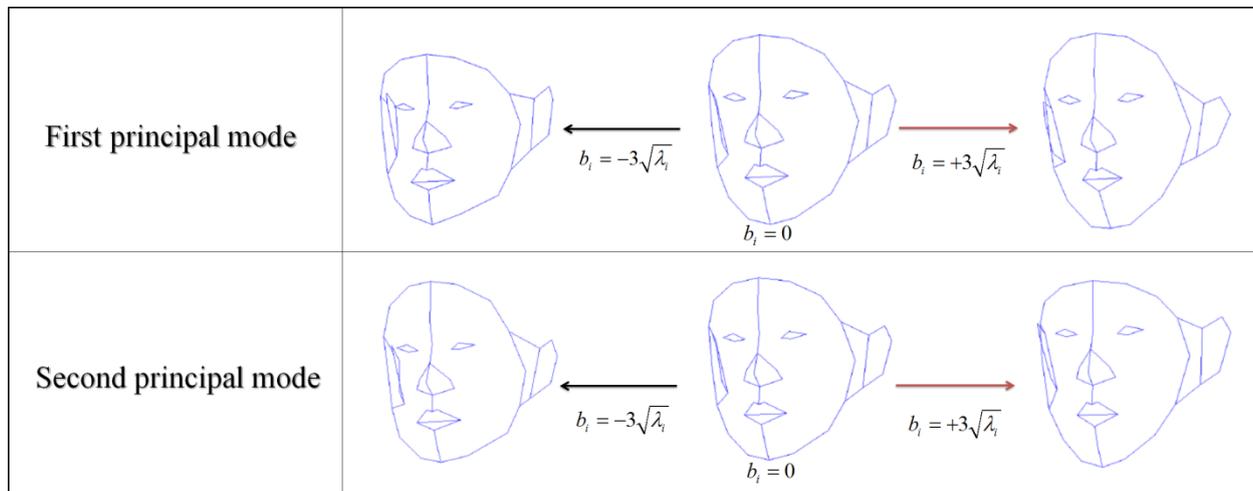
Then, the local deformation can be parameterized with model parameter (b) and the sample mean of 3D face shape data (\bar{S}). Finally, new 3D coordinates can be generated by the following equation:

$$\hat{S} = \bar{S} + \sum_{i=1}^m \Phi_i b_i \quad (6)$$

where m is the number of the modes of feature vectors.

In Equation (6), the local deformation of 3D FSM depends on only the model parameter (b) because the other variables are fixed. Therefore, determining the model parameter is a key to local deformations. Figure 1 shows deformation result of 3D FSM when the model parameters of first and second principal mode are changed.

Figure 1. 3D FSM deformation using model parameters of first and second principal mode. Deformation results applying a first principal mode parameter (**up**). Deformation results applying a second principal mode parameter (**bottom**).



3. 3D Facial Shape Estimation

In this section, we improve our proposed 3D face shape estimation method using a mirror system that was introduced at our previous work [29]. In the 3D face shape calculation using the 3D FSM, to use only the frontal face image may produce an inaccurate 3D face shape result that is different from the objective face shape because of uncertainty in the depth direction. To solve the problem, we calculate the 3D face shape by fitting multiple 3D FSMs to face images in three views. The multiple 3D FSMs include two virtual 3D FSMs which are applied to lateral face images and an original 3D FSM which is applied to a frontal face image. The virtual 3D FSMs can be generated by transforming the original 3D FSM symmetrically onto the mirror plane. After calculating the 3D face shape, detailed 3D face shapes can be interpolated by a generic 3D face model.

3.1. 3D Face Shape Estimation Using Multiple 3D FSMs in the Mirror System

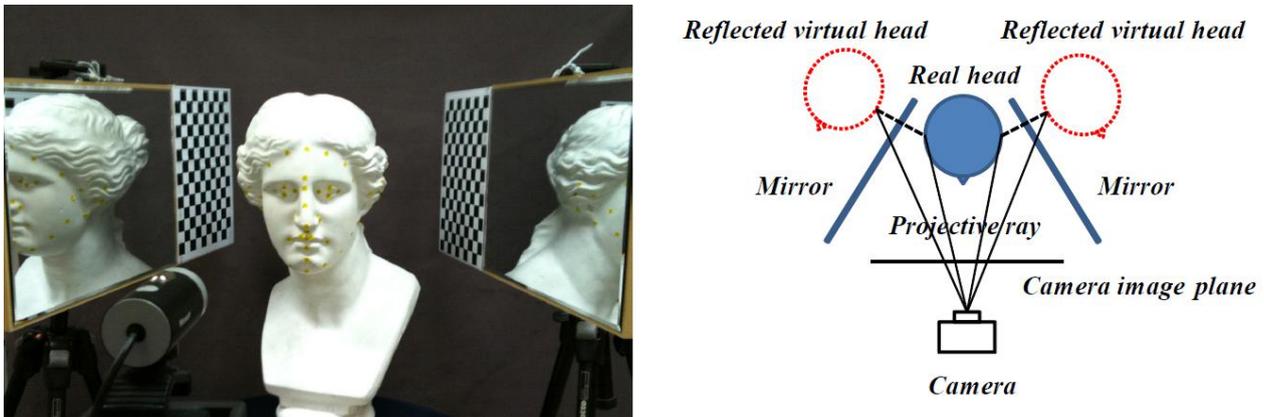
Our proposed face modeling system consists of two mirrors placed on either side of the face and a camera in front of the face. Frontal and lateral face images are captured simultaneously, and the pre-defined feature points are extracted from the captured image as described in Figure 2.

After feature extraction, the 3D FSM fitting procedure is carried out to calculate 3D coordinates from the extracted 2D feature points. During the fitting procedure, the 3D FSM parameters are adjusted to match the landmarks of the 3D FSM with the extracted feature points. This can be thought as least square optimization problem, and then the sum of the distances between the projected landmarks and objective feature points can be the cost function to be minimized. This cost function can be represented as:

$$\begin{aligned}
 F &= \|\mathbf{S}_{Obj} - \mathbf{S}_{ProjFSM}\|^2 \\
 \mathbf{S}_{Obj} &= [x_1, y_1, x_2, y_2, \dots, x_N, y_N]^T \\
 \mathbf{S}_{ProjFSM} &= [x'_1, y'_1, x'_2, y'_2, \dots, x'_N, y'_N]^T
 \end{aligned}
 \tag{7}$$

where N is the number of landmarks of 3D FSM, \mathbf{S}_{Obj} is the set of objective feature points in the image, and $\mathbf{S}_{ProjFSM}$ is the set of 3D FSM landmarks projected to the image plane.

Figure 2. A mirror system for capturing facial images and a conceptual diagram of the system.



The extracted feature points are categorized into three groups, \mathbf{S}_{FObj} , \mathbf{S}_{LObj} and \mathbf{S}_{RObj} depending on the direction of the face. Then, three cost functions can be generated by the following equations:

$$\begin{aligned}
 F_{Front} &= \|\mathbf{S}_{FObj} - \mathbf{S}_{FProjFSM}\|^2 \\
 F_{Left} &= \|\mathbf{S}_{LObj} - \mathbf{S}_{LProjFSM}\|^2 \\
 F_{Right} &= \|\mathbf{S}_{RObj} - \mathbf{S}_{RProjFSM}\|^2
 \end{aligned}
 \tag{8}$$

The total cost function is then the sum of these three variables:

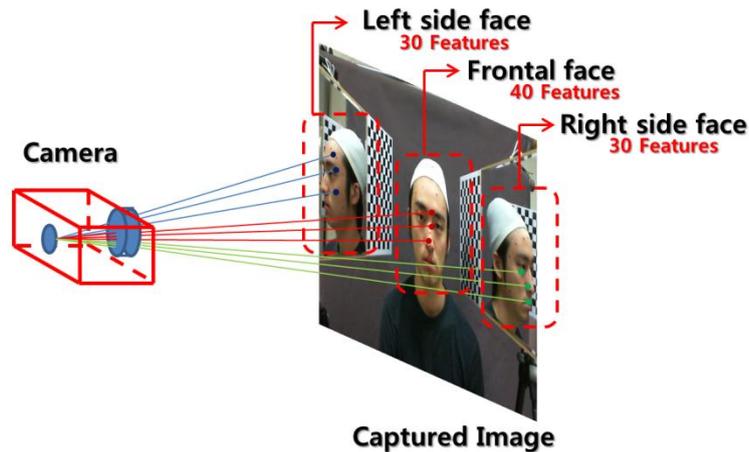
$$F_{Total} = F_{Front} + F_{Left} + F_{Right}
 \tag{9}$$

In practice, 30 left side face features are extracted, while features like the right ear, right eye, *etc.*, remain occluded. Then, a cost function for left side face is determined as described in Equation (8). Next, 30 right side face features are extracted while the left side features like left ear, left eye, *etc.* are occluded, and a cost function can be determined like in the left side face case. In the frontal face case, only features of the ears are occluded, so we extract 40 features as described in Figure 3.

Meanwhile, $\mathbf{S}_{FProjFSM}$ can be directly calculated from the perspective projection of 3D FSM in Equation (8). However, additional calculations with respect to the mirror reflection are required to acquire $\mathbf{S}_{LProjFSM}$ and $\mathbf{S}_{RProjFSM}$.

In mirror geometry, the mirror image of an object can be explained by the projection of a virtual 3D object reflected by the mirror plane onto the image plane, as described in Figure 2. A virtual 3D face can be generated by transforming the objective face symmetrically onto the mirror plane. Therefore, the location and orientation of the mirror planes should be calculated first.

Figure 3. Feature extraction in simultaneously captured images and cost function generation.



An ideal, perfectly flat mirror plane can be represented by:

$$aX + bY + cZ + d = 0 \quad (10)$$

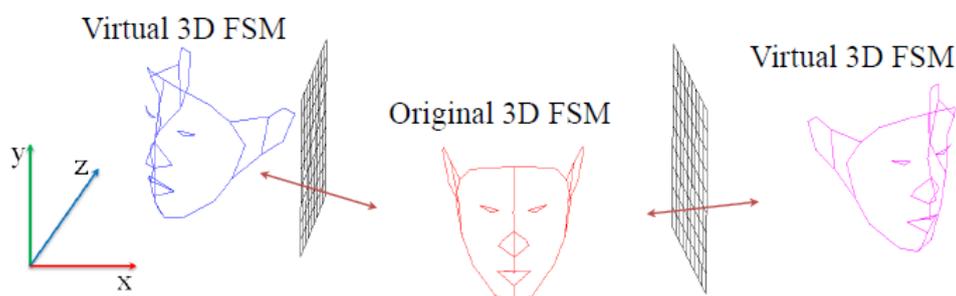
where (a,b,c) is the normal vector of the mirror plane, and (X,Y,Z) are the 3D coordinates of an arbitrary point on the mirror plane. Then, the unknown coefficients a,b,c and d can be easily calculated by solving a linear equation using singular value decomposition (SVD). The 3D points on the mirror plane can be obtained by Zhang's camera calibration method [30] after attaching checkerboards to the mirrors.

Once the plane equation is calculated, a virtual 3D face can be generated using a Householder reflection. Given the 3D FSM landmarks (\mathbf{P}_{real}) and the Householder matrix (\mathbf{H}_u), the virtual 3D face can be calculated by:

$$\begin{aligned} \mathbf{P}_{img} &= \mathbf{H}_u \mathbf{P}_{real} - \mathbf{H}_u \mathbf{P}_{plane} + \mathbf{P}_{plane} \\ \mathbf{H}_u &= (\mathbf{I}_{3 \times 3} - 2\mathbf{u}\mathbf{u}^T) \\ \mathbf{u} &= [A, B, C]^T \end{aligned} \quad (11)$$

where \mathbf{u} is the normal vector of the mirror plane, $\mathbf{I}_{3 \times 3}$ is the 3×3 identity matrix, and \mathbf{P}_{plane} is an arbitrary point on the mirror plane. Figure 4 describes the landmarks of 3D FSM and virtual 3D FSMs generated by the Householder reflection, where the red dots in the center represent original 3D FSM landmarks, and the green and blue dots represent the virtual 3D FSM landmarks derived from both mirrors. The red and green grids represent the mirror planes.

Figure 4. 3D FSM and two virtual 3D FSMs generated by a Householder reflection.



After generating the virtual 3D FSMs, we can calculate $\mathbf{S}_{LProjFSM}$ and $\mathbf{S}_{RProjFSM}$ by perspective projection under the camera coordinate system shown in Equation (12):

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & 0 & c_x \\ 0 & \alpha_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (12)$$

In Equation (12), α (α_x and α_y) is the scale parameter, and c (c_x and c_y) is the principal point with respect to the x and y axes. These parameters can be easily determined from the camera calibration result when we calculate the 3D points on the mirror plane using Zhang's method [30]. The second term on the left-hand side is the normalized perspective projection matrix.

After the total cost function (F_{total}) is determined, we calculate optimal solution to minimize it to fit the 3D FSM on the input face image. This can be expressed as:

$$\begin{aligned} \mathbf{x}^* &= \operatorname{argmin}_{\mathbf{x}} \{F_{total}(\mathbf{x})\} \\ \mathbf{x} &= [\hat{\mathbf{A}}, \mathbf{T}, \mathbf{b}]^T \end{aligned} \quad (13)$$

where $\hat{\mathbf{A}}$ is the column vector after stacking the elements of \mathbf{A} . In global deformation, the unknown parameters are elements of the matrices \mathbf{A} and \mathbf{T} , and the vector \mathbf{b} is an unknown parameter of local deformation.

To calculate the minimizer (\mathbf{x}^*), the iterative Levenberg-Marquardt optimization method is applied. In this method, in case of original 3D FSM, the partial derivatives of the Jacobian matrix can be easily calculated using the chain rule:

$$\begin{aligned} F &= \sum_{i=1}^n (f_i(x))^2, \quad \mathbf{f} = (\mathbf{S}_{Obj} - \mathbf{S}_{ProjFSM}) = [f_1 \quad \cdots \quad f_n] \\ \mathbf{J}(\mathbf{x}) &= \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_m} \end{bmatrix}, \quad \mathbf{x} = [\hat{\mathbf{A}}, \mathbf{T}, \mathbf{b}]^T \\ \frac{\partial f_j}{\partial A_i} &= \left(\frac{\partial f_j}{\partial (X_t, Y_t, Z_t)} \Big|_{P(X,Y,Z)} \right) \cdot \frac{\partial P(X,Y,Z)}{\partial A_i} \\ \frac{\partial f_j}{\partial T_i} &= \left(\frac{\partial f_j}{\partial (X_t, Y_t, Z_t)} \Big|_{P(X,Y,Z)} \right) \cdot \frac{\partial P(X,Y,Z)}{\partial T_i} \\ \frac{\partial f_j}{\partial b_i} &= \left(\frac{\partial f_j}{\partial (X_t, Y_t, Z_t)} \Big|_{P(X,Y,Z)} \right) \cdot \left(\frac{\partial P_t(X,Y,Z)}{\partial (X,Y,Z)} \Big|_{\sum_{i=1}^m S_i b_i} \right) \cdot \frac{\partial \sum_{i=1}^m S_i b_i}{\partial b_i} \end{aligned} \quad (14)$$

where m is the number of 3D FSM parameters, n is the dimension of residual \mathbf{f} .

However, for the virtual 3D FSM, the partial derivatives are changed due to the Householder transformation terms. After applying Equation (11) to Equation (1), the partial derivatives of the residual f can be calculated as:

$$\begin{aligned} \frac{\partial f_j}{\partial A_i} &= \left(\frac{\partial f_j}{\partial (X_{img}^k, Y_{img}^k, Z_{img}^k)} \Big|_{P_{img}} \right) \left(\frac{\partial P_{img}(X_t, Y_t, Z_t)}{\partial (X_t, Y_t, Z_t)} \Big|_P \right) \frac{\partial P(X, Y, Z)}{\partial A_i} \\ \frac{\partial f_j}{\partial T_i} &= \left(\frac{\partial f_j}{\partial (X_{img}, Y_{img}, Z_{img})} \Big|_{P_{img}} \right) \left(\frac{\partial P_{img}(X_t, Y_t, Z_t)}{\partial (X_t, Y_t, Z_t)} \Big|_P \right) \frac{\partial P(X, Y, Z)}{\partial T_i} \\ \frac{\partial f_j}{\partial b_i} &= \left(\frac{\partial f_j}{\partial (X_{img}, Y_{img}, Z_{img})} \Big|_{P_{img}} \right) \left(\frac{\partial P_{img}(X_t, Y_t, Z_t)}{\partial (X_t, Y_t, Z_t)} \Big|_P \right) \left(\frac{\partial P_t(X, Y, Z)}{\partial (X, Y, Z)} \Big|_P \right) \frac{\partial \sum_{i=1}^m S_i b_i}{\partial b_i} \end{aligned} \quad (15)$$

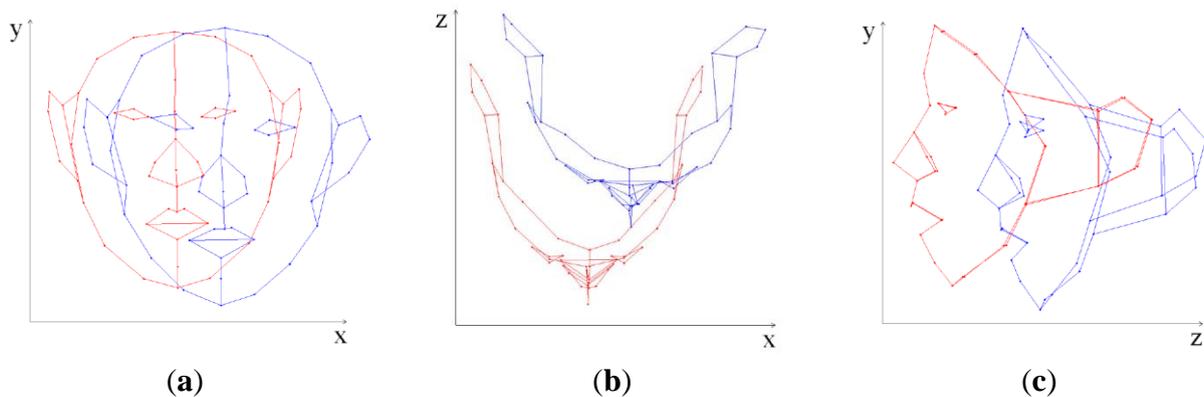
where A_i and T_i are the parameters of the 3D affine transformation, and b_i is a model parameter.

After calculating the Jacobian matrix about the three 3D FSM, the entire Jacobian matrix (\mathbf{J}_{All}) can be reformulated by stacking each individual matrix. Then, the gradient of the cost function (∇F) can be calculated by:

$$\begin{aligned} \mathbf{J}_{All} &= [\mathbf{J}_F, \mathbf{J}_{LS}, \mathbf{J}_{RS}]^T \\ \nabla F &= \mathbf{J}_{All}^T \mathbf{f} \end{aligned} \quad (16)$$

where \mathbf{J}_F is the Jacobian matrix from the front 3D FSM, \mathbf{J}_{LS} is the Jacobian matrix of the virtual 3D FSM from the left mirror, and \mathbf{J}_{RS} is the Jacobian matrix of the virtual 3D FSM from the right mirror. Figure 5 shows estimation results of the 3D FSM, where the blue dots represent the landmarks of the initial 3D FSM, and the red dots represent the landmarks of the 3D face shape estimated by our method.

Figure 5. The 3D face shape estimation results. (a) Frontal view, (b) bird eye's view, (c) side view. The blue dots represent the landmarks of the initial 3D FSM, and the red dots represent the landmarks of the 3D face shape estimated by our method.



3.2. Generic Model Fitting

After 3D face shape estimation, the 3D positions of other vertices can be determined by a generic 3D face model. The generic face model has been used in various applications because it has a uniform

point distribution and can provide detailed face shape with a small number of points [3,4,26,31]. Among the previous approaches, the most common is a deformation technique based on the radial basis function (RBF) which can deform the vertices of the generic model by establishing the deformation function between the estimated 3D face feature points and the corresponding points of the generic model. Generally, the deformation function $\mathbf{P}' = g(\mathbf{P})$ takes the form of low order polynomial terms \mathbf{M} and \mathbf{t} added to a weighted sum of the radial basis function $\phi(r)$ with constraints $\sum_i w_i = 0$ and $\sum_i w_i \mathbf{P}_i = 0$:

$$g(\mathbf{P}) = \sum_i w_i \phi(\|\mathbf{P} - \mathbf{P}_i\|) + \mathbf{M}\mathbf{P} + \mathbf{t} \quad (17)$$

In Equation (17), \mathbf{P}_i is a 3D feature point of the face, and \mathbf{P} is a corresponding vertex of the generic model. To determine the weights of the radial basis function w_i and the affine matrices \mathbf{M} and \mathbf{t} , Equation (18) is reformulated as linear equation $\Psi \mathbf{X} = \mathbf{Y}$, where Ψ , \mathbf{X} and \mathbf{Y} can be expressed as follows:

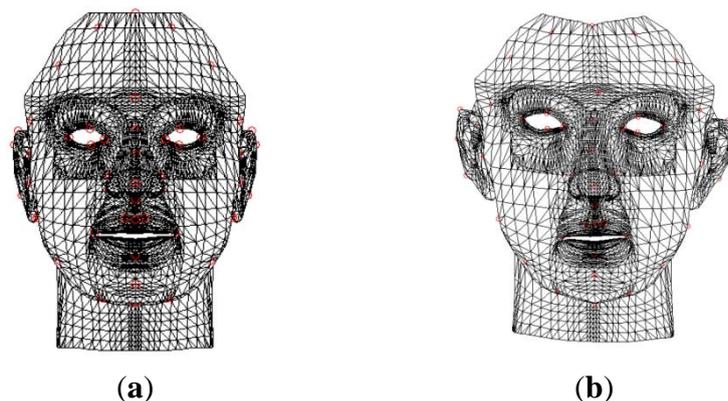
$$\Psi = \begin{bmatrix} \phi(\|\mathbf{p}_1 - \mathbf{p}_1\|) & \cdots & \phi(\|\mathbf{p}_1 - \mathbf{p}_n\|) & p_{1x} & p_{1y} & p_{1z} & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi(\|\mathbf{p}_n - \mathbf{p}_1\|) & \cdots & \phi(\|\mathbf{p}_n - \mathbf{p}_n\|) & p_{nx} & p_{ny} & p_{nz} & 1 \\ 1 & \cdots & 1 & 0 & 0 & 0 & 0 \\ p_{1x} & \cdots & p_{2x} & 0 & 0 & 0 & 0 \\ p_{1y} & \cdots & p_{2y} & 0 & 0 & 0 & 0 \\ p_{1z} & \cdots & p_{2z} & 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{(n+4) \times (n+4)} \quad (18)$$

$$\mathbf{X} = [w_1 \quad \cdots \quad w_n \quad \mathbf{M} \quad \mathbf{t}]^T \in \mathbb{R}^{(n+4) \times 3} \quad \mathbf{Y} = [\mathbf{q}_1 \quad \cdots \quad \mathbf{q}_n \quad 0 \quad 0 \quad 0 \quad 0]^T \in \mathbb{R}^{(n+4) \times 3}$$

Once all of the parameters of the deformation function are determined, the vertices of the generic model can be deformed by multiplying $\hat{\Psi}$ by \mathbf{X} . Similar to Ψ in Equation (18), $\hat{\Psi}$ can be established from other landmarks of the 3D FSM.

Figure 6 shows the initial generic face model and deformed generic model after RBF interpolation. We edited the generic model created by Pighin [5] for use in our system. In Figure 6, the red circles in the generic model represent the 3D shape landmarks estimated in Section 2.2.

Figure 6. (a) Initial generic model and (b) deformed generic model after RBF interpolation.



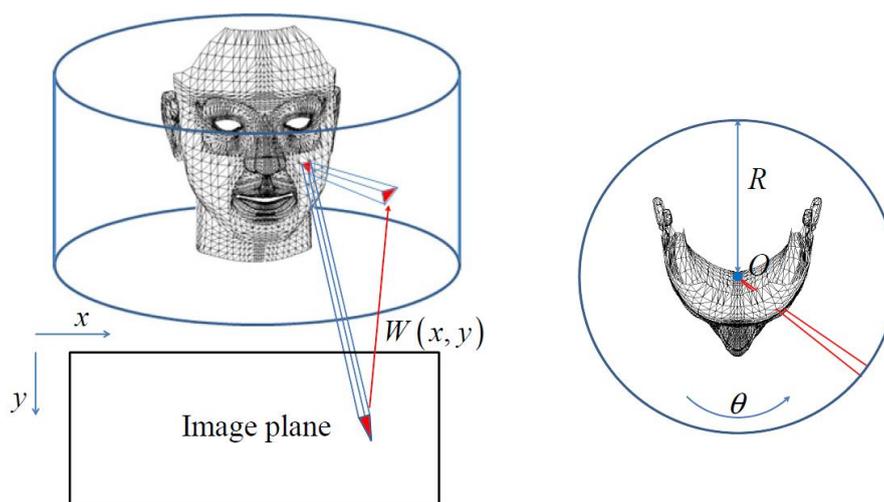
4. Texture Map Generation and Mapping

In this section, we introduce cylindrical mapping and address the stitching method using a seamless cloning method. A seam appears at the boundaries of each face part because of photometric inconsistency after stitching. To solve this problem, a seamless cloning method [15] with a gradient-domain blending technique is applied. This method successfully removed the seams at the boundaries, and thus a texture map of the whole face could be created.

4.1. Texture Extraction Using Cylindrical Texture Mapping

To map textures on the 3D face model, a texture map is created by extracting the texture directly from the captured face image. For the sake of simplicity, cylindrical mapping is applied. In common cylindrical mapping methods, mesh vertices that are intersected with the ray passing through the center of a cylinder are projected onto the image plane after a virtual cylinder is placed around the 3D face model. Then, the colors of the corresponding pixels in the image are extracted and mapped to the texture map. However, this is time consuming because the positions of the vertices on the face mesh must be calculated. Thus, in our texture mapping procedure, vertices of a triangle mesh are projected onto the image plane, and then textures in the projected mesh are warped on the texture map, as shown Figure 7. Our system stitches together texture maps from the three different views to create a texture map of the entire face.

Figure 7. Cylindrical texture mapping for facial texture extraction.

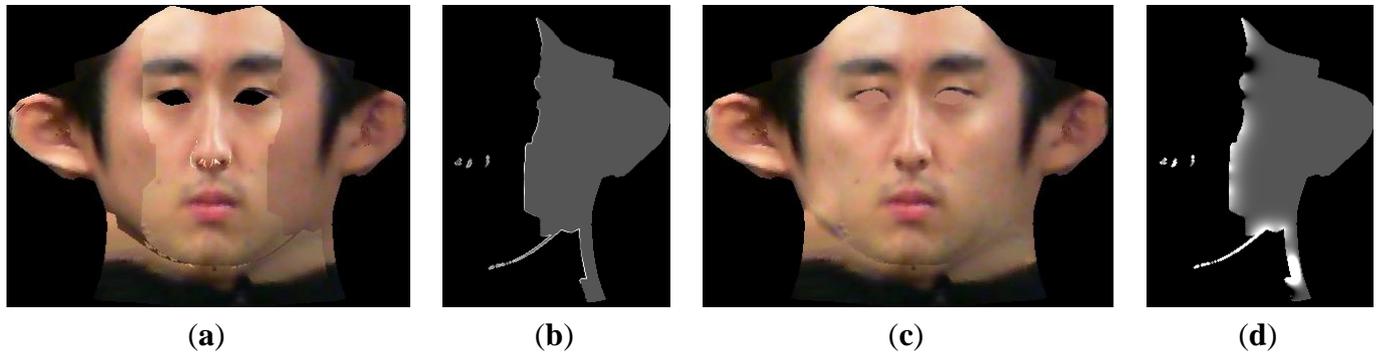


4.2. Texture Map Generation Using Modified Image Stitching Method

As addressed in Section 4.1, a texture map of the entire face can be created by stitching each face texture parts. However, a seam appears at the boundaries of each face parts because of photometric inconsistency, as described in Figure 8(a). To solve this problem, a seamless image stitching method with gradient-domain blending techniques [15] is applied. Generally, gradient-domain blending techniques are more efficient at reducing photometric inconsistencies than is general image-domain blending. To implement the seamless cloning method, the entire texture image is first divided into three parts along the boundaries. Then, overlapping regions are created by expanding both parts by

1 pixel at the encountering region. Next, a $h(x,y)$ map is created that is the same size as the original texture map of the whole face.

Figure 8. Texture map refinement using seamless cloning. (a) Face texture map after merging each face texture part, (b) initial map for seamless cloning, (c) synthesized face texture map after seamless cloning (d) $h(x,y)$ map after 100 iterations of Equation (12).



For initialization of the $h(x,y)$ map, the side facial region having 0 pixel values is changed to have 1 pixel values. The pixel values in the overlapping regions are set to the ratio of the pixel value of the front view (g) to the pixel value of the side view (f):

$$h(x,y) = \begin{cases} 1 & g(x,y) = 0 \\ & \text{or } (x,y) \in R \\ \frac{f(x,y)}{g(x,y)} & \text{otherwise} \end{cases} \quad (19)$$

where R is the side region.

The $h(x,y)$ value is iteratively updated by calculating the solution of Laplace's equation at the corresponding pixel:

$$h^{n+1}(x,y) = \frac{1}{4} (h^n(x+1,y) + h^n(x-1,y) + h^n(x,y+1) + h^n(x,y-1)) \quad (20)$$

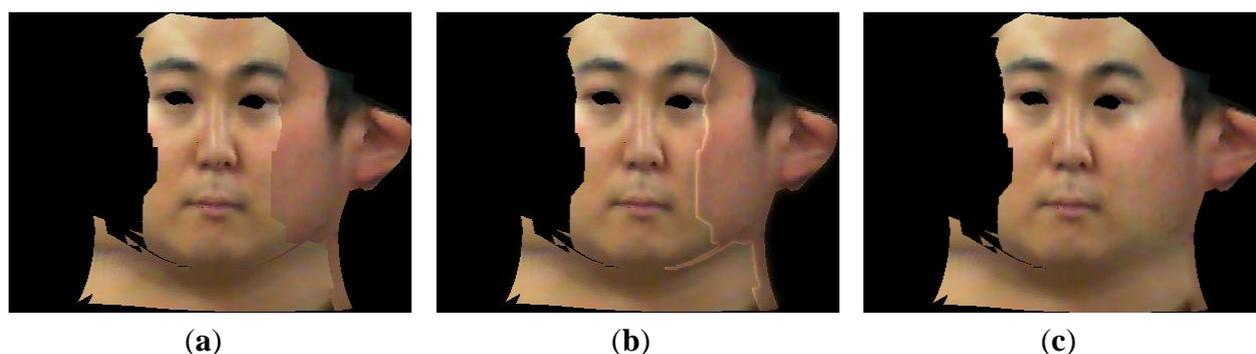
Figure 8(b,d) shows an initial $h(x,y)$ map and the resulting $h(x,y)$ map after 100 iterations. In Figure 8(b), the $h(x,y)$ values at the boundary are propagated into the valid area. After sufficient iterations, the final pixel values on the side of the face are calculated as the product of the updated $h(x,y)$ and original pixel value, which allows for creation of a continuous texture map at the boundary:

$$\hat{f}(x,y) = h^n(x,y) \cdot g(x,y) \quad (21)$$

After using morphological operations to fill in holes, a final face texture map can be created, as shown in Figure 8(c). Figure 9 shows the final 3D face after refining the texture map and the addition of artificial eyes.

Figure 9. 3D face modeling result after texture mapping.

Generally, multi-resolution splining [32] is well known as a blending method that operates well when the overlapped regions between images to be splined are broad enough. In our system, the overlapped region between each face part is only one pixel wide, therefore we couldn't get a satisfactory result when using multi-resolution splining method as described in Figure 10(b). On the other hand, the gradient domain image stitching method that is applied to our system shows excellent performance in spite of the narrow overlapped region as described in Figure 10(c). Figure 10 shows that the seam between boundaries of texture parts as described in Figure 10(a) is completely removed, while the seam remains after applying multi-resolution splining.

Figure 10. Image stitching results using multi-resolution splining and our method. (a) Image stitching without any blending methods. (b) Image stitching using multi-resolution splining. (c) Image stitching using our method.

5. Experiments and Results

5.1. Experimental Settings

Before constructing the proposed face modeling system, we completed statistical analyses with respect to the 3D scan landmarks in order to calculate the feature vector elements of the local deformation parameters in the FSM, as described in Section 2.1. For the statistical analysis, we used principal component analysis (PCA). We recorded 3D face views of 100 individuals with a *CyberwareTM* laser scanner and then selected 50 landmarks in each face. During the PCA procedure, we retained 90% of the eigenvalue energy spectrum to reduce computational complexity.

To define the ground truth, we captured 3D faces of 30 individual with a 3D laser scanner at the same time that we captured the image with our proposed system. We attached color markers on each user's face to identify the feature points as shown in Figure 11 and then manually measured their 3D coordinates.

Figure 11. The user's face captured with our mirror system. The red color markers are attached on user's face for performance test.



For a relative comparison, we used Lin's method [9]. That work is a good reference to evaluate the performance of our method because their mirror-based face modeling system is similar to ours. Additionally, we can indirectly compare our method and ordinary 3D reconstruction methods using epipolar geometry because they already compared their work with ordinary 3D reconstruction methods using epipolar geometry.

5.2. Accuracy Tests

We implemented Lin's method [22] and calculated the 3D coordinates of the marked feature points. In Lin's method, only the 3D coordinates of visible features can be reconstructed, so we compared the 3D reconstruction results of the 40 features on the frontal face with the actual faces. To compare the results with the actual faces, we aligned the 3D points from both methods with the 3D points of the actual faces. 3D procrustes analysis [28] was used to align the 3D coordinates of the reconstructed points. After aligning the 3D points, we calculated the average sum of the Euclidean distances between each point on the reconstruction and the actual faces, as described in Equation (22):

$$E = \frac{1}{M} \sum_{i=1}^M \|S_{Reconst} - S_{Ground}\|^2$$

$$S_{Reconst} = [x_1, y_1, z_1, x_2, y_2, z_2 \cdots, x_N, y_N, z_N]^T$$

$$S_{Ground} = [x'_1, y'_1, z'_1, x'_2, y'_2, z'_2 \cdots, x'_N, y'_N, z'_N]^T$$
(22)

Then, we compared the accuracy of their method with that of our method, as shown in Table 1, where our proposed method exhibits slightly higher absolute errors than that of Lin, but the standard deviation (Std) of our method was about two times lower.

Table 1. Mean, standard deviation and median of absolute distance errors of the proposed and Lin. I-C's method compared to the actual faces.

Estimation Method	Absolute Error (mm)		
	Mean	Std	Median
Lin.'s method [22]	3.12	1.14	2.59
Proposed method	3.58	0.59	3.49

5.3. Test on Robustness to Feature Extraction Error

To test on the robustness of our method with respect to feature extraction errors, we artificially generated erroneous feature points with normally distributed random distances and directions. Firstly, we calculated a two-dimensional matrix containing normally distributed random numbers using the Box-Muller method. Then, we generate the noisy feature points by adding each column vector of the matrix to the 2D coordinates of the feature points in the input face image. The feature points on the face for measurement were annotated by color markers. We assumed that the 2D positions of the marked feature points were the reference position.

We first tested the results of our proposed method and Lin's method according to error strength, which can be adjusted by changing the standard deviation of the random numbers. Table 2 shows the maximum error distances according to the standard deviation.

Table 2. The maximum error distances according to the standard deviation.

Standard deviation of the error	1	2	3	4	5
Maximum error distances (pixel)	3.811	6.692	9.760	12.667	16.845

We carried out the 3D face shape reconstruction by applying the proposed method and Lin's method with noisy feature points. The standard deviation of the error was varied from 0 to 5 in intervals of 0.02. Then, we calculated the average sum of the Euclidean distances (average absolute error) between each 3D reconstruction point and the truth. As shown in Figure 12(a), the average absolute error of the proposed method increases monotonically, while the average absolute error of Lin's method increases and fluctuates much more rapidly.

Next, we fixed the standard deviation and measured the average absolute error as the number of noisy feature points was increased from 0 to 100 in intervals of 1. As shown in Figure 12(b), the results are similar to those of the first robustness test. The average absolute error of the proposed method increases monotonically, but the average absolute error of Lin's method increases and fluctuates much more rapidly.

Figure 13 shows the 3D face modeling results of proposed method and that of Lin with erroneous feature points. The standard deviation of the error is fixed at 3. As shown in Figure 13, the result of Lin's method shows a significantly distorted shape near the erroneous feature points. On the other hand, the proposed method maintains the overall face shape, even with the noisy feature points.

Figure 12. Robustness test results. (a) Error measurement results with increasing error strength by changing standard deviation of normally distributed random numbers. (b) Error measurement results with increasing number of noisy feature points.

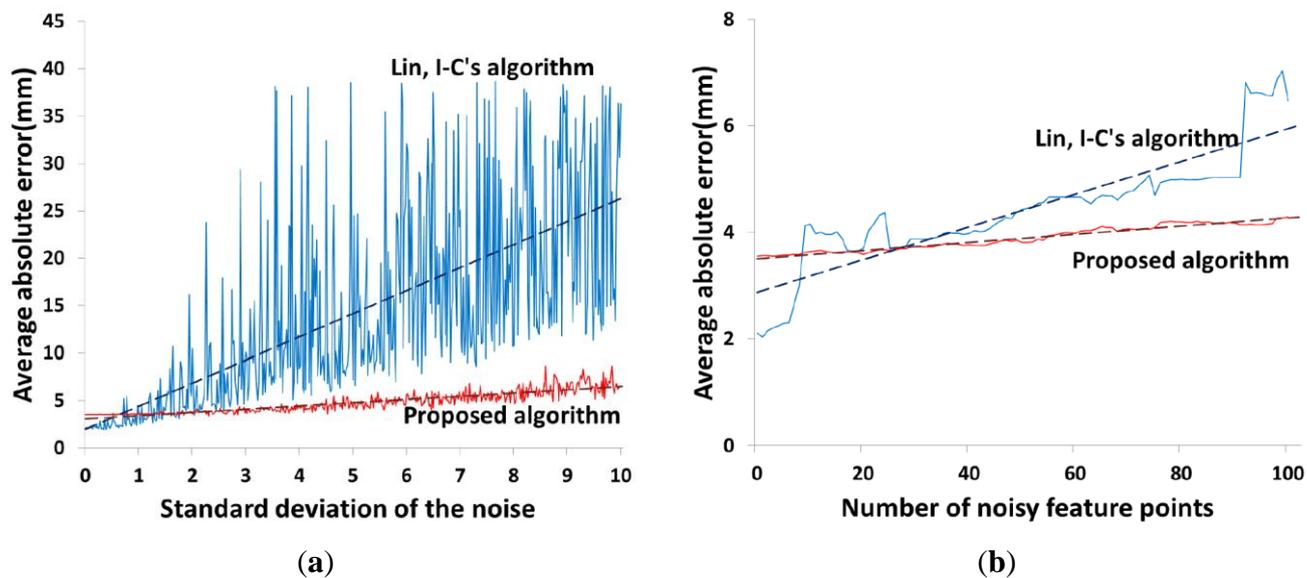
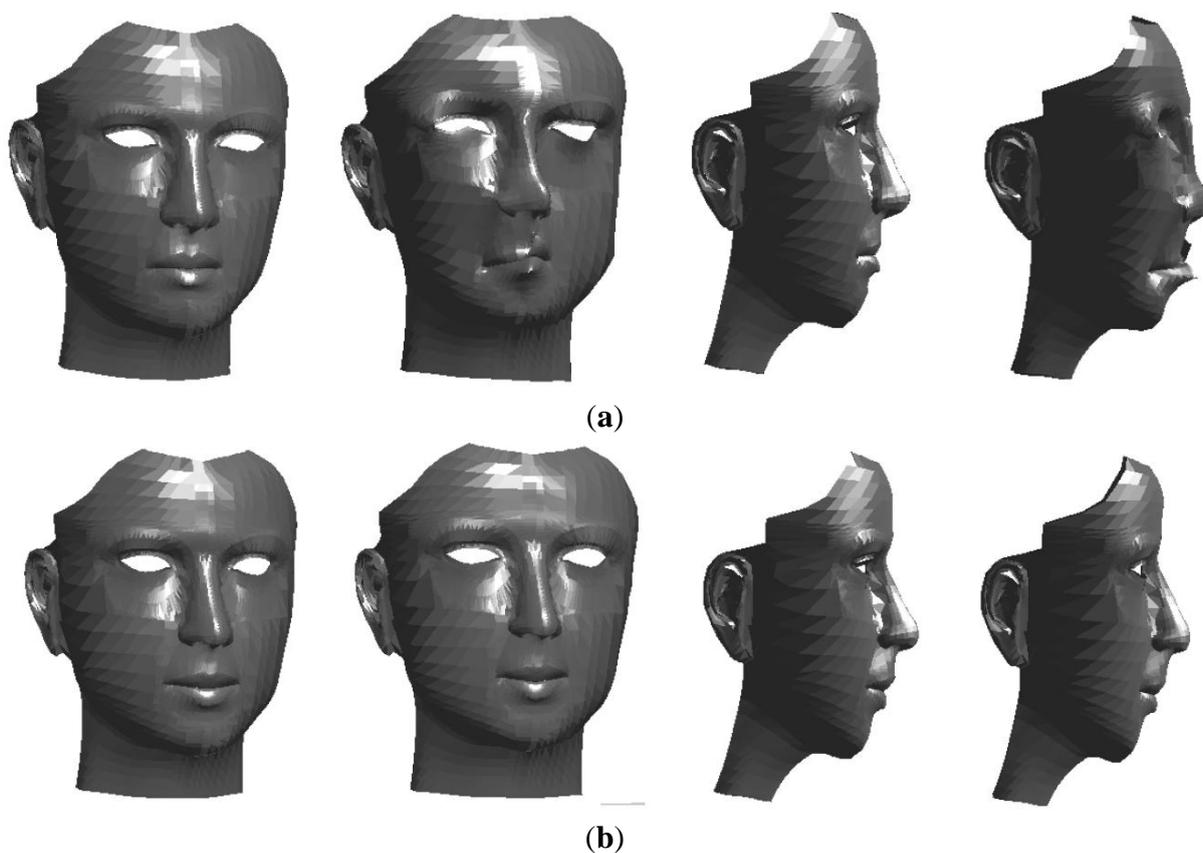


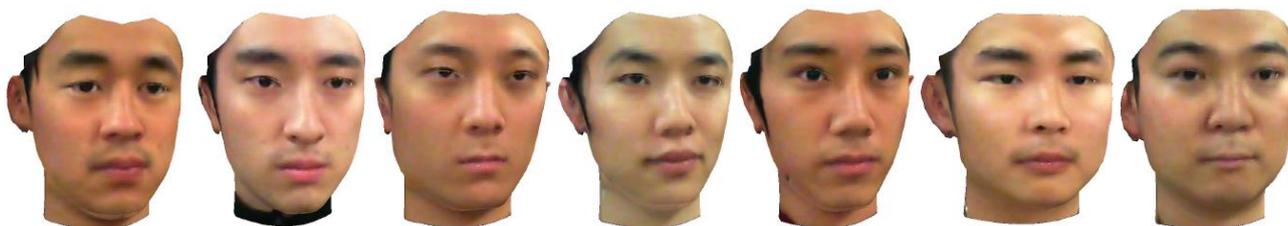
Figure 13. (a) 3D facial shape estimation results of Lin's method. (b) Proposed method with erroneous feature points.



5.4. Textured 3D Face Model Generation Results

We generated a textured 3D face model of users using the proposed face modeling method. After applying the generic model fitting as described in Subsection 3.2, we applied our texture mapping method described in Section 4. Eyeballs are not included in the generic model, and so we inserted artificial eyeballs with the 3D Max program. After producing the eyeballs, we align the center of the eyeball to the center of the eye region. Figure 14 shows the input face image, the texture map and the textured 3D face.

Figure 14. The results of the textured 3D face model for individuals.



6. Conclusions

In this paper, we propose a realistic 3D face modeling method that is robust to feature extraction errors and can generate accurate 3D face models. In the facial shape estimation procedure, we propose a 3D face shape estimation method using multiple 3D face deformable models in a mirror system. The proposed method shows high robustness to feature extraction errors and highly accurate 3D face modeling results, as described in Sections 5.2 and 5.3. In the texture mapping procedure, we apply cylindrical mapping and stitching technique to generate a texture map. We apply the seamless cloning method, which is a type of gradient-domain blending technique, to remove the seam that caused by photometric inconsistency and finally can thus acquire a natural texture map.

To evaluate our method's performance, we carry out accuracy and robustness tests with respect to 30 individuals' 3D facial shape estimation results. Our method shows not only highly accurate 3D face shape results when compared with the ground truth, but also robustness to feature extraction errors. Moreover, the 3D face rendering results intuitively show that our method is more robust to feature extraction errors than other 3D face shape estimation methods. An additional contribution of our method is that wide range of face textures can be acquired by the mirror system. Lastly, we generate textured 3D faces using our proposed method. The results show that our method can generate very realistic 3D faces, as shown in Figure 14. Our ultimate goal is to create an automatic 3D face modeling system, and so we plan to apply automatic feature extraction processes to our method. This may be a problem for side view images and will require the development of new techniques that will be described in future works.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0011472). This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2011-0016302).

References

1. Ersotelos, N.; Dong, F. Building highly realistic facial modeling and animation: A survey. *Vis. Comput.* **2008**, *24*, 13–30.
2. Sansoni, G.; Trebeschi, M.; Docchio, F. State-of-the-art and applications of 3D imaging sensors in industry, cultural heritage, medicine, and criminal investigation. *Sensors* **2009**, *9*, 568–601.
3. Remondino, F.; El-Hakim, S. Image-based 3D modeling: A review. *Photogram. Rec.* **2006**, *21*, 269–291.
4. Lee, Y.; Terzopoulos, D.; Waters, K. Realistic Modeling for Facial Animation. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, Los Angeles, CA, USA, 6–11 August 1995; pp. 55–62.
5. Yu, Z.; Edmond, C.P.; Eric, S. Constructing a realistic face model of an individual for expression animation. *Int. J. Inform. Technol.* **2002**, *8*, 10–25.
6. Crocombe, A.D.; Linney, A.D.; Campos, J.; Richards, R. Non-contact anthropometry using projected laser line distortion: Three-dimensional graphic visualization and applications. *Opt. Lasers Eng.* **1997**, *28*, 137–155.
7. Fua, P. From multiple stereo views to multiple 3D surfaces. *Int. J. Comput. Vis.* **1997**, *24*, 19–35.
8. Ku, A. Implementation of 3D optical scanning technology for automotive applications. *Sensors* **2009**, *9*, 1967–1979.
9. Gonzalez-Aguilera, D.; Gomez-Lahoz, J.; Sanchez, J. A new approach for structural monitoring of large dams with a three-dimensional laser scanner. *Sensors* **2008**, *8*, 5866–5883.
10. Hartley, R.I.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2004.
11. Blanz, V.; Vetter, T. A Morphable Model for the Synthesis of 3D Faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, Los Angeles, CA, USA, 8–13 August 1999; pp. 187–194.
12. Cootes, T.; Edwards, G.; Taylor, C. Active Appearance Models. In *Proceedings of the 5th European Conference on Computer Vision*, Freiburg, Germany, 2–6 June 1998; pp. 484–498.
13. Cootes, T.; Taylor, C. Constrained Active Appearance Models. In *Proceedings of the 8th IEEE International Conference on Computer Vision*, Vancouver, BC, Canada, 7–14 July 2001; pp. 748–754.
14. Cootes, T.; Taylor, C.; Cooper, D.; Graham, J. Active shape models—Their training and application. *Comput. Vis. Image Understand.* **1995**, *61*, 38–59.
15. Georgiev, T. Covariant Derivatives and Vision. In *Proceedings of the 9th European Conference on Computer Vision*, Graz, Austria, 7–13 May 2006; pp. 56–69.
16. Lee, W.S.; Magnenat-Thalmann, N. Generating a Population of Animated Faces from Pictures. In *Proceedings of IEEE International Workshop on Modelling People*, Kerkyra, Greece, 29 September 1999; pp. 62–69.
17. Akimoto, T.; Suenaga, Y.; Wallace, R. Automatic creation of 3D facial models. *IEEE Comput. Graph. Appl.* **1993**, *13*, 16–22.
18. Ip, H.H.S.; Yin, L. Constructing a 3D individualized head model from two orthogonal views. *Vis. Comput.* **1996**, *12*, 254–266.

19. Fua, P. Regularized Bundle-adjustment to model heads from image sequences with calibration data. *Int. J. Comput. Vis.* **2000**, *38*, 153–171.
20. Lee, T.Y.; Lin, P.H.; Yang, T.H. Photo-Realistic 3D Head Modeling Using Multi-View Images. In *Computational Science and Its Applications—ICCSA 2004*; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3044, pp. 713–720.
21. Pighin, F.; Hecker, J.; Lischinski, D.; Szeliski, R.; Salesin, D. Synthesizing realistic facial expressions from photographs. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, Orlando, FL, USA, 19–24 July 1998; pp. 75–84.
22. Lin, I.; Yeh, J.; Ouhyoung, M. Extracting 3D facial animation parameters from multi-view video clips. *IEEE Comput. Graph. Appl.* **2002**, *22*, 72–80.
23. Blanz, V.; Scherbaum, K.; Seidel, H.P. Fitting a Morphable Model to 3D Scans of Faces. In *Proceedings of the 11th IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, 14–20 October 2007; pp. 1–8.
24. Kuo, C.; Huang, R.; Lin, T. Synthesizing Lateral Face from Frontal Facial Image Using Anthropometric Estimation. In *Proceedings of International Conference on Image Processing*, Washington, DC, USA, 26–29 October 1997; pp. 133–136.
25. Kuo, C.; Huang, R.S.; Lin, T.G. 3-D facial model estimation from single front-view facial image. *IEEE Trans. Circ. Syst. Video Technol.* **2002**, *12*, 183–192.
26. Baek, S.; Kim, B.; Lee, K. 3D Face Model Reconstruction from Single 2D Frontal Image. In *Proceedings of the 8th International Conference on Virtual Reality Continuum and Its Applications in Industry*, Yokohama, Japan, 14–15 December 2009; pp. 95–101.
27. Blanz, V.; Vetter, T. A Morphable Model for the Synthesis of 3D Faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, Los Angeles, CA, USA, 8–13 August 1999; pp. 187–194.
28. Ansari, A.; Abdel-Mottaleb, M. 3D Face Modeling Using Two Views and A Generic Face Model with Application to 3D Face Recognition. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, Miami, FL, USA, 21–22 July 2003; pp. 37–44.
29. Hwang, J.; Kim, W.; Ban, Y.; Lee, S. Robust 3D Face Shape Estimation Using Multiple Deformable Models. In *Proceedings of the 6th IEEE Conference on Industrial Electronics and Applications*, Miami, FL, USA, 21–23 June 2011, pp.1953–1958.
30. Zhang, Z. Flexible Camera Calibration by Viewing a Plane from Unknown Orientations. In *Proceedings of the 7th IEEE International Conference on Computer Vision*, Kerkyra, Greece, 20–27 September 1999; pp. 666–673.
31. Widanagamaachchi, W.; Dharmaratne, A. 3D Face Reconstruction from 2D images. In *Proceedings of 2010 International Conference on Digital Image Computing: Techniques and Applications*, Canberra, ACT, Australia, 1–3 December 2008, pp. 365–371.
32. Lee, W.-S.; Thalmann, N.M. Fast head modeling for animation. *J. Image Vis. Comput.* **2000**, *18*, 355–364.