Efficient 3D Model based Face Representation and Recognition Algorithm using Pixel-to-Vertex Map (PVM)

Kanghun Jeong and Hyeonjoon Moon
Department of Computer Science and Engineering, Sejong University
98 Gunja-Dong, Gwangjin-Gu, Seoul, 143-747 Republic of Korea
[e-mail: hmoon@sejong.ac.kr]
*Corresponding author: Hyeonjoon Moon

Received July 12, 2010; revised Sepember 22, 2010; revised November 6, 2010; revised December 20, 2010; accepted January 4, 2010; published January 31, 2011

Abstract

A 3D model based approach for a face representation and recognition algorithm has been investigated as a robust solution for pose and illumination variation. Since a generative 3D face model consists of a large number of vertices, a 3D model based face recognition system is generally inefficient in computation time and complexity. In this paper, we propose a novel 3D face representation algorithm based on a pixel to vertex map (PVM) to optimize the number of vertices. We explore shape and texture coefficient vectors of the 3D model by fitting it to an input face using inverse compositional image alignment (ICIA) to evaluate face recognition performance. Experimental results show that the proposed face representation and recognition algorithm is efficient in computation time while maintaining reasonable accuracy.

Keywords: Face recognition, face representation, vertices optimization, optical flow, image registration, 3D morphable model

DOI: 10.3837/tiis.2011.01.013

1. Introduction

Face recognition technology can be used in a wide range of applications such as identity authentication, access control, and surveillance systems. A face recognition system should be able to deal with various changes in face images. However, the variations between the images of the same face due to illumination and head pose are almost always larger than image variation due to change in face identity. Traditional face recognition systems have primarily relied on 2D images. However, they tend to give a higher degree of recognition performance only when images are of good quality and the acquisition process can be tightly controlled.

Recently, a large amount of literature on 3D based face recognition has been published with various methods and experiments. 3D has several advantages over traditional 2D face recognition: First, 3D data provides absolute geometrical shape and size information of a face. Additionally, face recognition using 3D data is more robust to pose and posture changes since the model can be rotated to any arbitrary position. Also, 3D face recognition can be less sensitive to illumination since it does not solely depend on pixel intensity for calculating facial similarity. Finally, it provides automatic face segmentation information since the background is typically not synthesized in the reconstruction process [1]. V. Blanz, T. Vetter and S. Romdhani [2][3] proposed a method using a 3D morphable model for face recognition robust to pose and illumination. They constructed a 3D morphable model with 3D faces acquired from a 3D laser scanner, which was positioned at well calibrated cylindrical coordinates. The texture information of the 3D face was especially well defined in the reference frame where one pixel corresponds to one 3D vertex perfectly. Thereafter, they found appropriate shape and texture coefficient vectors of the model by fitting it to an input face using Stochastic Newton Optimization (SNO) [3] or Inverse Compositional Image Alignment (ICIA) [4][5] as a fitting algorithm and then evaluated the face recognition performance of a collected 3D face database based on Pose In Emotion (PIE) [6], FacE REcognition Test (FERET) [7] and the Face Recognition Vendor Test (FRVT) [8]. However, this approach has complexity and inefficiency problems caused by very large vertex number (about 80,000 ~ 110,000) despite excellent performance.

In this paper we propose a novel 3D face representation algorithm based on a pixel to vertex map (PVM) and optical flow on texture (face image). It is possible to reduce the vertex number and it can be aligned with correspondence information of a 3D reference face simultaneously. This paper is organized as follows. The next section presents the proposed algorithm for 3D face representation. Section 3 simply describes the procedure of fitting the 3D model to an input image. Experimental results are presented in Section 4 based on a Korean database. Finally, conclusions and future work are discussed in Section 5.

2. Face Representation

In general, a 3D face scan consists of texture intensity in 2D frame and geometrical shape in 3D. Specifically, the shape information is represented by close connections of many vertices and polygons. And texture intensity in a 2D frame connects to vertex points by 1:N mapping. Since the vertex number of each of the 3D face scans is different from the others, it is necessary to manipulate them to have the same number of vertices and remove texture illumination for consistent mathematical expression and drawing in scene [9][10]. We propose a novel 3D face representation algorithm for vertex number correspondence, which is

performed by masking, pixel to vertex map (PVM), optical flow and alignment as showed in **Fig. 1** Zou et al. [26] proposed a landmark based face representation algorithm from a range image. Our proposed algorithm has optimized the face representation method based on a pixel to vertex map (PVM) which minimizes computational complexity while maintaining reasonable recognition performance.

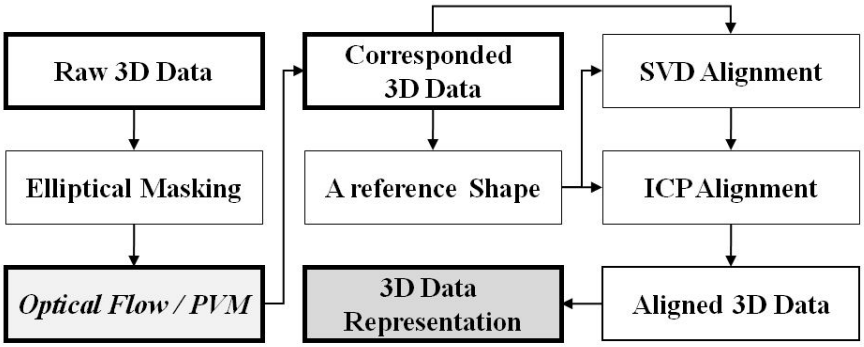

Fig. 1. 3D Face representation process.

2.1 3D Face Database

3D faces are collected using a Geometrix Face Vision 200, which is a stereo-camera based capturing device offering a 3D face scan including approximately 30,000 ~ 120,000 vertices and corresponding 2D texture images. There are 218 3D face scans collected from 110 people. A "session" is defined as collecting one set of 3D face data for each person with the same pose and lighting condition. There were three sessions each with a two week time period in our 3D face database. 3D face data per person has been used for model generation and experiments were performed with respect to the frontal poses. We have three sessions in our database with frontal views and neutral expressions only. We used 100 scans in session 1, for 3D model generation and 52 scans in other sessions for the performance evaluation. Also, 7 profile face images with a range from 15 to 40 degrees were acquired separately using the same device for variant pose test [11]. **Fig. 2** shows our collected face database.

2.2 Texture Correspondence using the Optical Flow

A 3D face scan is expressed with 2D texture information including the pixel intensities and 3D shape information including polygons. And it is a well-known method for model generation in model-based face recognition systems to adopt linear combinations of some face samples. But simple linear combination of some faces can be exposed to a critical problem unless they utilize correspondence information based on shape and texture. Texture data is different from pose and lighting conditions which are addressed through preprocessing.

In [22] they use albedo to exploit the fact that the set of images of an object are in a fixed pose (It is a convex cone in the space of images). Albedo is defined as the ratio of total-reflected to incident electromagnetic radiation. It is a unitless measure indicative of a surface's or body's diffuse reflectivity [24]. Using a small number of training images of each face taken with different lighting directions, the shape and albedo of the face can be reconstructed. This reconstruction serves as a generative model that can be used to render-or synthesize- images of the face under novel poses and illumination conditions. The pose space is then sampled and the corresponding illumination cone is approximated by a low

dimensional linear subspace. This method performs almost without error, except on the most extreme lighting directions, and significantly outperforms popular recognition methods that do not use a generative model [22].

Fig. 2. Face database.

Albedo assumes single point light source but in real world situation, actual lighting condition is extremely complex with multiple point light sources. Additionally, their experiment [22] assumes under variable illumination with a fixed pose, our experiment is focused on solving pose correction and projects onto 2D face space for recognition with fixed illumination. Our 3D face database has uniform illumination condition since it is collected under a controlled environment.

While image alignment has been studied in different areas of computer vision for decades, aligning images depicting different scenes remains a challenging problem. Scale Invariant Feature Transform (SIFT) flow is analogous to optical flow where an image is aligned to its temporally adjacent frame. The SIFT flow algorithm consists of matching densely sampled, pixel wise SIFT features between two images, while preserving spatial discontinuities. Although derived from optical flow, SIFT flow is drastically different from optical flow. In SIFT flow, correspondence is built upon pixel wise SIFT features instead of RGB or gradient that was used in optical flow. Optical flow is formulated in a discrete manner, the search window size in SIFT flow is much larger than that in optical flow. SIFT flow utilizes a larger window compared to optical flow since an object can move drastically from one image to another in scene alignment [23]. In our experiment, we used the same reference face data with histogram equalization for face region (**Fig. 3**).

Fig. 3. Preprocessed face data with hitogram equalization.

In our experiment, optical flow and SIFT flow have been examined. The experimental results of the output flow of both algorithms and face images are shown in **Fig. 4**. As shown in

Fig. 4-(b), the characteristics of the data for the optical flow information was obtained for the eyes, nose and mouth area.

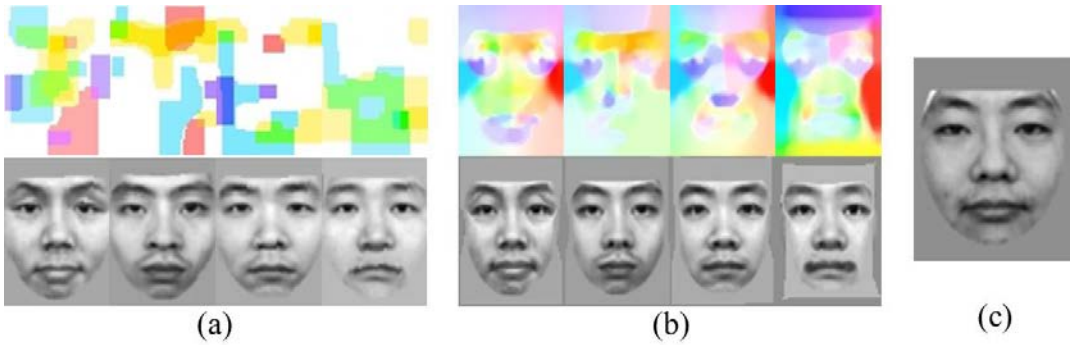

Fig. 4. Result of SIFT flow and optical flow. **(a)** top images: flow data of SIFT flow, bottom images: result of SIFT flow **(b)** top images: flow data of optical flow, bottom images: result of optical flow **(c)** reference image.

Table 1. Result of optical flow and SIFT flow.

File_ID	Optical flow		SIFT flow	
	Elapsed Time (sec)	Max flow	Elapsed Time (sec)	Max flow
1	0.410662	3.6233	0.526346	3
2	0.417441	3.0577	0.457788	2.2361
3	0.417858	3.0577	0.470041	2
4	0.415745	3.1779	0.469789	2.8284
5	0.41677	5.6345	0.459024	3
6	0.419404	3.0536	0.492103	3.1623
7	0.414255	4.5795	0.468067	4.1231
8	0.421285	5.881	0.467489	3.1623
9	0.414271	5.4978	0.470509	3.1623
10	0.418041	3.2479	0.459031	3.1623
Mean Time (sec)	0.4162472	3.86782	0.4656161	3.48069

As shown in **Table 1**, output flow data produces different results in SIFT and optical flow. Our experimental results shows that the face data for the optical flow performs slightly better performance in computation time and classification of face region. Typically, SIFT flow's performance is better than optical flow for general object images. However, our experimental result shows that optical flow produces better performance since our database consists of static face images only. In **Fig. 2**, there is a sample face database which is used for this experiment. Based on the static face database, we generate the average 3D face model and the input face image is recognized through the 2D-3D projection and fitting process. Therefore, the average 3D face model is the standard data to extract features from and to compare characteristic for the input image. The 3D model is utilized to perform the recognition process in time and space domain regardless of the facial image region and system registered face data. It is suitable to

use SIFT flow for the real-time face recognition system with either the motion or time varying face image database.

In this experiment, we choose optical flow to eliminate the image bluring problem and to achieve texture correspondence in our experiment. Optical flow is the distribution of apparent velocities of movement of brightness patterns in an image. Optical flow can arise from relative motion of objects and the viewer. Optical flow can give important information about the spatial arrangement of the objects viewed and the rate of change of this arrangement [12]. Originally, optical flow **was** a technique to analyze moving patterns of an object. Based on the work presented by Horn and Schunck [13], we perform the texture correspondence under the following assumption.

There are N+1 faces in the database and the first face image is selected as a reference face. A fundamental idea for utilizing the optical flow in our system is that each of the remaining faces is regarded as a time-delayed version of the reference. Let the face image intensity at the point (x, y) in the image plane at time be denoted by $E(x, y, t)$. Now consider what happens when the face moves. Theoretically, the intensity of a particular point in the face is constant regardless of time consumed, so that differentiation with respect to time t is zero. Using the chain rule for differentiation, we are sure that

$$\frac{dE}{dt} = \frac{\partial E}{\partial x}\frac{dx}{dt} + \frac{\partial E}{\partial y}\frac{dy}{dt} + \frac{\partial E}{\partial t} \tag{1}$$

For compact notation, if we let the velocity components in x and y directions be

$$E_x u + E_y \upsilon = -E_t \tag{2}$$

where $E_x, E_y,$ and E_t are the partial derivatives of pixel intensity with respect to $x, y,$ and t respectively. To determine the component of the movement u, υ explicitly, an additional condition must be required. We are able to derive a constraint from the fact that neighboring points in the rigid object such as the face have nearly similar velocities.

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \qquad \nabla^2 \upsilon = \frac{\partial^2 \upsilon}{\partial x^2} + \frac{\partial^2 \upsilon}{\partial y^2} = 0 \tag{3}$$

Now, we must compute the estimated solutions of the two constraints (see (2) and (3)). In the first place, consider the derivatives of intensity using the discrete property of image in **Fig. 5**, we consider a cube formed by eight measurements [13]. Each of the estimates the average of the four first differences taken over adjacent measurements in the cube as defined in (4), (5) and (6).

$$\begin{aligned} E_x \approx \frac{1}{4}[&(E_{i,j+1,k} - E_{i,j,k}) + (E_{i+1,j+1,k} - E_{i+1,j,k}) + \\ &(E_{i,j+1,k+1} - E_{i,j,k+1}) + (E_{i+1,j+1,k+1} - E_{i+1,j,k+1})] \end{aligned} \tag{4}$$

$$E_x \approx \frac{1}{4}[(E_{i+1,j,k} - E_{i,j,k}) + (E_{i+1,j+1,k} - E_{i,j+1,k}) + (E_{i+1,j,k+1} - E_{i,j,k+1}) + (E_{i+1,j+1,k+1} - E_{i,j+1,k+1})] \quad (5)$$

$$E_x \approx \frac{1}{4}[(E_{i,j,k+1} - E_{i,j,k}) + (E_{i+1,j,k+1} - E_{i+1,j,k}) + (E_{i,j+1,k+1} - E_{i,j+1,k}) + (E_{i+1,j+1,k+1} - E_{i+1,j+1,k})] \quad (6)$$

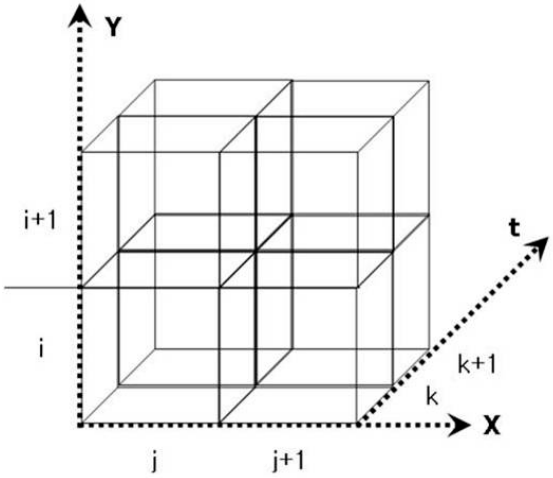

Fig. 5. A cube formed by eight measurements.

Secondly, we need to estimate the Laplacians defined in (3). A useful approximation takes the following form

$$\nabla^2 u \approx 3(\hat{u}_{i,j,k} - u_{i,j,k}), \qquad \nabla^2 \upsilon \approx 3(\hat{\upsilon}_{i,j,k} - \upsilon_{i,j,k}) \quad (7)$$

Where the estimated local derivatives $\hat{u}$ and $\hat{\upsilon}$ are defined by a 3×3 mask, which assigns the weights to the 8-neighbor points as shown in **Fig. 6-(b)**.

$$\hat{u}_{i,j,k} = \frac{1}{6}(u_{i-1,j,k} + u_{i,j+1,k} + u_{i+1,j,k} + u_{i,j-1,k}) + \frac{1}{12}(u_{i-1,j-1,k} + u_{i-1,j+1,k} + u_{i+1,j+1,k} + u_{i+1,j-1,k}) \quad (8)$$

$$\hat{\upsilon}_{i,j,k} = \frac{1}{6}(\upsilon_{i-1,j,k} + \upsilon_{i,j+1,k} + \upsilon_{i+1,j,k} + \upsilon_{i,j-1,k}) + \frac{1}{12}(\upsilon_{i-1,j-1,k} + \upsilon_{i-1,j+1,k} + \upsilon_{i+1,j+1,k} + \upsilon_{i+1,j-1,k}) \quad (9)$$

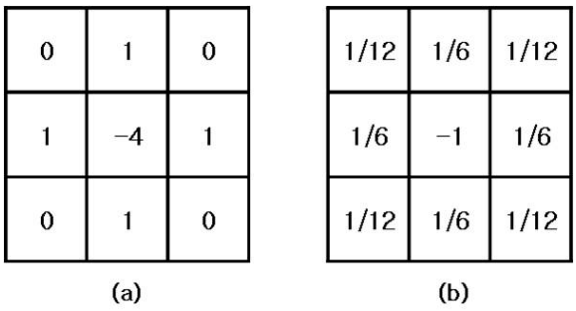

Fig. 6. Laplacian masks: **(a)** general **(b)** proposed.

At this point, the remaining problem is to minimize the cost function for the rate of change of image intensity. From two constraints defined previously, the cost function for total error is taken by the following forms.

$$C = \iint (\alpha^2 \Xi_L^2 + \Xi_G^2) dxdy \tag{10}$$

$$E_G = E_x u + E_y \upsilon + E_t \tag{11}$$

$$E_L^2 = (\partial u/\partial x)^2 + (\partial u/\partial y)^2 + (\partial \upsilon/\partial x)^2 + (\partial \upsilon/\partial y)^2 \tag{12}$$

Using the calculus of variation, we obtain

$$E_x^2 u + E_x E_y \upsilon = \alpha^2 \nabla^2 u - E_x E_t \tag{13}$$

$$E_x E_y u + E_y^2 \upsilon = \alpha^2 \nabla^2 \upsilon - E_y E_t \tag{14}$$

By substituting the estimation of the Laplacians defined in (8) and (9), (13) and (14) are modified to a simultaneous linear equation form as in the following

$$\begin{bmatrix} \alpha^2 + E_x^2 & E_x E_y \\ E_x E_y & \alpha^2 + E_y^2 \end{bmatrix} \begin{bmatrix} u \\ \upsilon \end{bmatrix} = \begin{bmatrix} \alpha^2 \hat{u} - E_x E_t \\ \alpha^2 \hat{\upsilon} - E_y E_t \end{bmatrix} \tag{15}$$

Finally, a solution (u, υ) for optical flow of a position in face image can be found using an iterative method such as the Gauss-Seidel method such that

$$u^{n+1} = \hat{u}^n - E_x [E_x \hat{u}^n + E_y \hat{\upsilon}^n + E_t] / (\alpha^2 + E_x^2 + E_y^2) \tag{16}$$

$$\upsilon^{n+1} = \hat{\upsilon}^n - E_y [E_x \hat{u}^n + E_y \hat{\upsilon}^n + E_t] / (\alpha^2 + E_x^2 + E_y^2) \tag{17}$$

The modified examples by texture correspondence using optical flow specifically described in the earlier part are shown in **Fig. 7**. The images in the first and third columns are the masked images and those in the second and fourth columns are the texture corresponded images.

Fig. 8 shows the result conducting the texture correspondence with ten face images. As shown in **Fig. 8-(c)**, there are a few blurred regions, especially, around the important features, such as eyes, eyebrows, nose and mouth. The average face is defined as a mean value of raw texture information for different faces. **Fig. 8-(a)** is generated before utilizing texture

correspondence. The blurring effect is remarkably decreased in **Fig. 8-(d)** since we utilize texture correspondence by the optical flow algorithm.

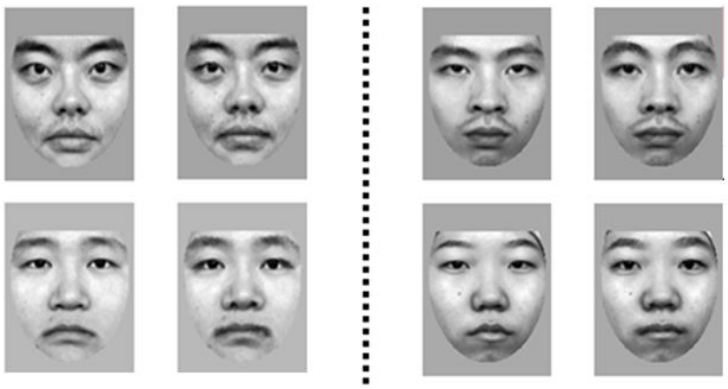

Fig. 7. Result of texture correspondence using optical flow.

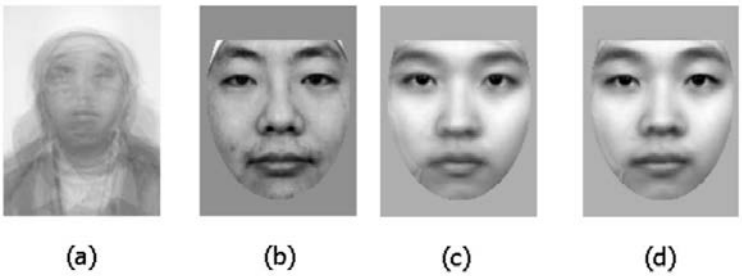

Fig. 8. The result conducting the texture correspondence with 10 face samples. **(a)** average face of raw texture value. **(b)** reference image for texture correspondence. **(c)** average face before texture correspondence. **(d)** average face after texture correspondence.

2.3 Pixel to Vertex Map (PVM)

In the first place, we have registered three fiducial points, which are left eye, right eye and mouth center (from the 2D texture information of each 3D face scan). Thereafter, a fixed point for each of three fiducial points is set in a texture frame. Each fiducial point of the 3D face scans is located in the same position of the texture frame. Then, a face region is preprocessed and separated from the background by adopting an elliptical mask (**Fig. 9**).

A face region is separated by following steps. First, we translate the center of the eyes of the destination to the center of its 2D texture information. Second, we must seek a suitable scale and rotation matrix. The first and second diagonal terms of the scale matrix means the scale variation in the horizontal and vertical directions, respectively. They can be simply computed based on a suitable relation on eyes and mouth positions of the destination and source.

A rotation matrix only depends on a rotation angle to the camera axis (for convenience, it is designated to z-axis). The rotation angle can be simply determined as the slope between two lines connecting both of the eyes of the destination and source 2D texture information. Finally, we translate the center of the present 2D texture data to the center of the eyes in source 2D texture data.

Then, it is noted that the signs of the displacements in the source translation matrix are opposite to those of in the destination translation matrix. These steps allow us to possess the

face images distributed in the same horizontal range. Additionally, it is required to perform a normalization scheme in the vertical direction for more reliable preprocessing. This process utilizes the correspondence of the vertical distance between the mouth and eyes.

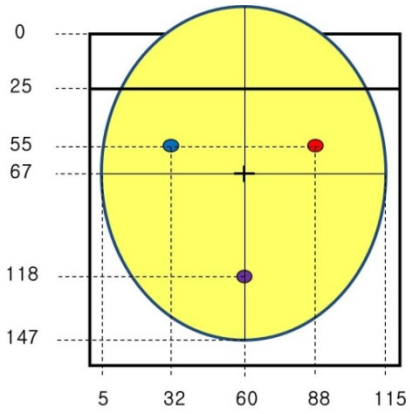

Fig. 9. Proposed elliptical mask.

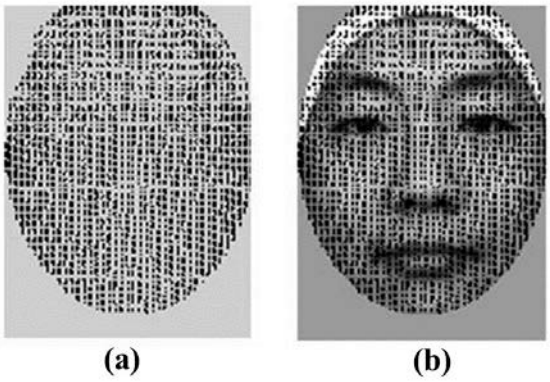

Fig. 10. An example showing a pixel-to-vertex map(PVM). AP and IP are expressed as dark and bright pixels in an elliptical mask, respectively. **(a)** Former active pixel. **(b)** Former active pixel + texture data.

A pixel-to-vertex map (PVM) is a sort of binary image, which classifies pixels in the masked face region into ones mapped to a vertex and the opposite. We call the former active pixel (AP) and the latter inactive pixel (IP). A PVM example is shown in **Fig. 10**.

The procedures for the vertex correspondence using a PVM are as follows:

- Construct each PVM matrix of M+1 3D face scans and build the vertex position matrix by stacking the position vector of the vertex mapping to each AP in a PVM. If the resolution of the texture frame is C by R, the PVM matrix of the *i*-th scan, denoted by $\mathbf{M}_i$ and the vertex position matrix of the *i*-th scan, denoted by $\mathbf{P}_i$ are obtained as

$$\mathbf{M}_i = \begin{bmatrix} m_{11}^i & m_{12}^i & \cdots & m_{1c}^i \\ m_{21}^i & & & \\ \vdots & & \ddots & \vdots \\ m_{R1}^i & & \cdots & m_{RC}^i \end{bmatrix}, \qquad m_{rc} = \begin{cases} 0, & if \quad p_{rc} \; is \quad \mathbf{IP} \\ 1, & if \quad p_{rc} \; is \quad \mathbf{AP} \end{cases} \tag{18}$$

$$\mathbf{P}_i = \begin{bmatrix} \mathbf{V}_1^i & \mathbf{V}_2^i & \cdots & \mathbf{V}_{s(\mathbf{M}_i)}^i \end{bmatrix} \tag{19}$$

  where p_{rc} is the pixel positioned at (r,c) in the texture frame and s($\mathbf{M}_i$) is size of the PVM, the number of the APs in the PVM. Also, $\mathbf{V}_j^i = \begin{bmatrix} x_j & y_j & z_j \end{bmatrix}^T$ is the 3D position vector of the vertex mapping to the j^{th} AP in the i^{th} scan.
- Select a reference pixel-to-vertex map (RPVM), denoted by $\mathbf{M}^R$, by maximizing this criterion.

$$\mathrm{M}^R = \arg\max_{\mathrm{M}_i} s(\mathrm{M}_i) \tag{20}$$

The size of the RPVM, $s(\mathbf{M}^R)$ means the vertex number of a reduced subset. Then, all scans will be in correspondence with the vertex number. Likewise, the vertex position matrix of the RPVM is denoted by $\mathbf{P}^R$.

- Compute each modified vertex position matrix of all scans except one selected for the RPVM.

$$\hat{\mathbf{V}}_k^i = \begin{cases} \mathbf{V}_{p(k)}^i & if \;\; m_{p(k)}^i \;\; is \;\; \mathbf{AP} \\ \mathbf{V}^N & if \;\; m_{p(k)}^i \;\; is \;\; \mathbf{IP} \end{cases} \tag{21}$$

$$\mathbf{V}^N = \sum_{q=1}^{8} W_q \mathbf{V}_q \tag{22}$$

Where $\hat{\mathbf{V}}_k^i$ is a modified vertex position vector, which is in the same position as that of the original vertex if mapped to AP, otherwise it should be acquired by an interpolation method. And, the subscripted $\mathbf{P}(k)$ means the position of the pixel mapped to the vertex related to *k*-th column in the $\mathbf{P}^R$. We have to seek an appropriate 3D position $\mathbf{V}^N$ for a vertex mapped to IP using linear combinations of the positions of vertices mapped to the 8 nearest neighbor APs in the PVM of the target scan as defined in (22).

2.4 Face Alignment and Model Generation

Through the PVM presented in the previous subsection, it is possible that all 3D face scans in our database (see Section 4) are expressed with the same number of vertex points. To construct a more accurate model, it is necessary to utilize some techniques for face alignment, which is transforming the geometrical factors (scale, rotation angles and displacements) of a target face based on a reference face. Face alignment in our research is achieved by adopting singular value decomposition (SVD) [14] and Iterative Closest Points (ICP) [15] sequentially. The dimension of the vector space for a 3D face sample is expressed in a matrix form. 3D face samples can be expressed by corresponding vertex numbers. Then, we apply Principal Component Analysis (PCA) [16] to make an eigenspace for 3D shapes and textures. PCA is a statistical method of reducing the dimension of a vector space in terms of the distribution of variance. Additionally, we explore Linear Discriminant Analysis (LDA) which converts high dimensional image space into a significantly lower dimensional feature space which is insensitive both to variation in pose and lighting condition. In LDA, feature vectors are recalculated to minimize inter-class distance while maximizing intra-class distance for better separation of the classes than PCA. Therefore, we constructed separate models from shapes and textures of 100 Korean people by applying PCA [17] and LDA [18] independently. The separate models are generated by linear combination of the shapes and textures as given by (23).

$$\mathbf{S} = \mathbf{S}_0 + \sum_{j=1}^{N_S} \alpha_j \mathbf{S}_j, \qquad \mathbf{T} = \mathbf{T}_0 + \sum_{j=1}^{N_T} \beta_j \mathbf{T}_j \tag{23}$$

where $\alpha = [\alpha_1 \ \alpha_2 \ \cdots \ \alpha_{N_S}]$ and $\beta = [\beta_1 \ \beta_2 \ \cdots \ \beta_{N_S}]$ are the shape and texture coefficient vectors (should be estimated by a fitting procedure). Also, $\mathbf{S}_0$ and $\mathbf{S}_j$ are the shape average models and the eigenvectors associated with the *j*-th largest eigenvalue of the shape covariance matrix, $\mathbf{T}_0$ and $\mathbf{T}_j$ in textures likewise.

3. Fitting the Face Model using Inverse Compositional Image Alignment

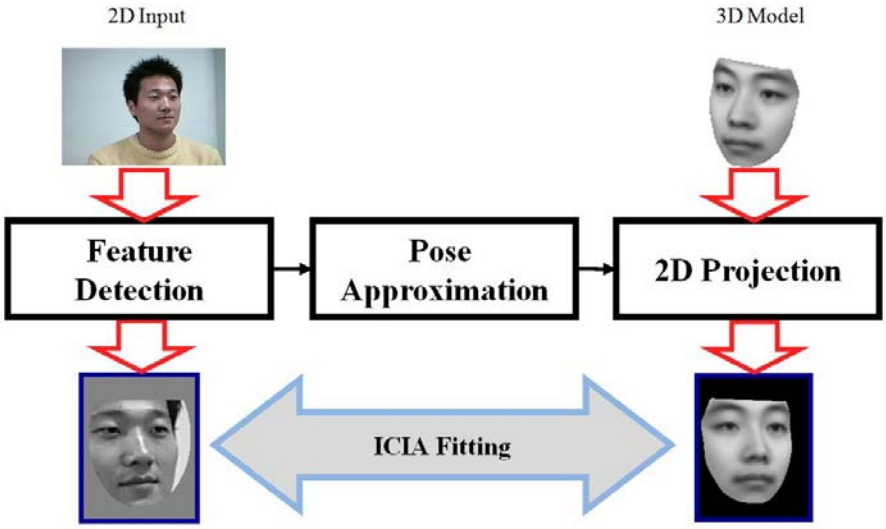

Fig. 11. ICIA fitting process.

Shape and texture coefficients of the generative 3D model are estimated by fitting it to a given input face. This is performed iteratively as close as possible to the input face. Fitting algorithms, called stochastic Newton optimization (SNO) and inverse compositional image alignment (ICIA) were utilized in [3]. It is generally accepted that SNO is more accurate but computationally expensive and ICIA is less accurate but more efficient in computation time [4]. We also explore the ICIA algorithm as a fitting method to guarantee the computational efficiency. Given an input image, initial coefficients of shape and texture and projection parameters for the model are selected appropriately. Initial coefficients of shape and texture usually have zero values but projection parameters are manually decided by the registration of some important features. Then, fitting steps are iterated until they converge to a given threshold value, minimizing the texture difference between the projected model image and the input image.

During the fitting process, texture coefficients are updated without an additive algorithm for each iteration. But in the case of shape coefficients, their updated values are not acquired with ease because of the nonlinear problem of structure from motion (SFM) [19]. To solve it, we recover the shape coefficients using the SVD based global approach [20] after the convergence (**Fig. 11**, **Fig. 12**).

Fig. 12. ICIA fitting result. The images in the top row are input images and those in the bottom row are the fitted versions of the 3D model. The inputs to the third column are frontal and the others are rotated 30 degrees approximately.

4. Experimental results

4.1 Experiment configuration

The performance evaluation of the proposed system was conducted in an identification scenario. In the identification task, when a face image of an unknown person is given to the system, the unknown face image is compared to a database of known people. It is then assumed that the individual in the unknown image is in the known database. There are three types of data sets required for the performance evaluation of the face recognition system [21]. One is a set for the purpose of generating a face model or training a face space, another is a set composed of data that must be beforehand enrolled in the system, which will be used as exemplars for recognition, and the other is a set composed of images which claim identity authentication to the recognition system. The second and third sets are frequently referred to as the gallery set and probe set, respectively. We used two databases. One is composed of 3D face scans collected by a stereo-camera based sensing device and the other is composed of 2D face images acquired from a web camera.

Table 2. Combinatorial configurations of gallery and probe sets.

Only Frontal Pose (A)		
Index	**Gallery Set**	**Probe Set**
A-1	3D samples in session 2	3D samples in session 3
A-2	2D images in a session	2D frontal images in a session
Pose Variation (B)		
Index	**Gallery Set**	**Probe Set**
B	2D frontal images in each session	All 2D profile images

In the first place, 100 people's 3D face data captured in the first session are exclusively used for only 3D model generation. The residual frontal data in the 3D database and all the images in the 2D database are utilized as candidates of the gallery and probe sets divided by an appropriate policy as listed in **Table 2**. A remarkable fact here is that all the candidates of the gallery set are frontal. The reason for this is that the frontal pose is generally required for user

enrollment in practical applications. The conducted experiments for performance evaluation are categorized into two cases according to whether data in the probe set are frontal or profile. First, we conducted two tests on the frontal case. Next, two tests were performed on the profile case. We will leave more detail descriptions of this configurations, in addition to their experimental results, for the next section. All data used in our experiment are in 120×160 resolution.

4.2 Test result on probes with frontal pose

We generated a 3D face model using a PVM representation algorithm and evaluation for a 3D face recognition algorithm on collected 3D face databases based on FRVT [8]. **Fig. 13** is a generated 3D model by the PVM representation algorithm using 100 Korean people face database. A 3D face model is used for face recognition with various pose and lighting conditions. The experimental results show the recognition rate for frontal and profile face images.

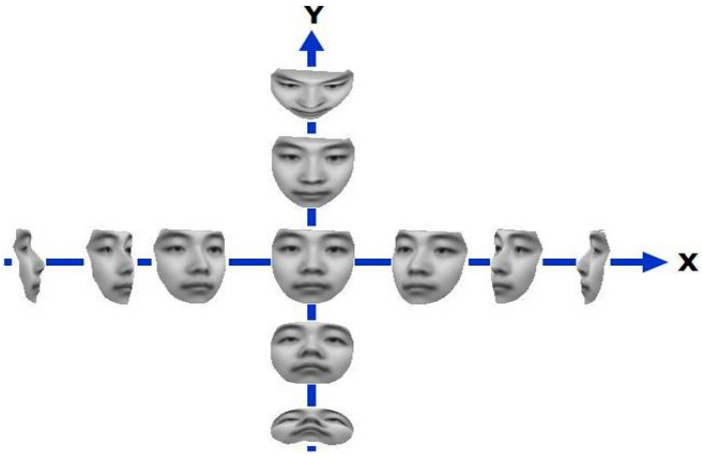

Fig. 13. Generated 3D face model.

As can be ascertained in **Table 2**, two tests on images with frontal pose were conducted by changing the combinations of the gallery and probe sets. Combination A-1 is the most ideal combination because the gallery and probe sets are selected from texture images acquired in the same environment to 100 exemplar face scans used for 3D face model generation. For this combination, we utilized 62 samples collected in the third session of the 3D face database as candidates of the probe set, and for the gallery set 62 samples out of 92 samples from the second session, of which each identity is accurately matched with that of each probe.

Table 2. The results on cumulative recognition rate to rank 10 in A-1.

Rank & Fitting Time	Hit count			Recognition rate (%)		
	Texture	Shape	Both	Texture	Shape	Both
1	50	23	50	80.7	37.1	80.7
2	55	31	55	88.7	50.0	88.7
3	56	35	57	90.3	56.5	91.9
4	57	38	58	91.9	61.3	93.6

5	58	39	59	93.6	62.9	95.2
6	62	41	62	100.0	66.1	100.0
7	62	45	62	100.0	72.6	100.0
8	62	47	62	100.0	75.8	100.0
9	62	50	62	100.0	80.7	100.0
10	62	50	62	100.0	80.7	100.0
Mean Time (sec)	0.9	5.6	5.6			

We achieved three tests on this combination. The purpose of the first test was to confirm the influences of shape and texture parameters used as an identity parameter. When only texture parameters, only shape parameters, and both parameters are used as the identity parameter, respectively, the experimental results are shown in **Table 3**. The rank is defined as the possibility of the top N matches among candidates [25]. We have provided mean fitting time since there are no variations among different ranks as shown in **Table 3**. For the convenience of explanation, the cases are denoted by 'Texture', 'Shape', and 'Both', respectively. In this test, we selected shape and texture basis of the generative model as 50 percent of total face feature vectors It can be verified that the recognition rate in rank 1 is 80.7% (50 hits out of total 62) in 'Texture' and 'Both' cases.

Also, the accuracy of both the cases concurrently converged to 100% (62 out of 62) in rank 6. Note that the accuracy rate in the 'Shape' case is very remarkably low against the other cases. For this reason, the experimental data is dominant on texture data compared to the shape data. Shape data requires more iteration than texture data in order to calculate feature parameter but contributes less in recognition performance (**Table 3**).

Based on our experimental results, it is probably conferred that the texture is a more important factor than shape for classification of identity. On the other hand, average fitting time per an image of 'texture' and 'both' case is 0.9s and 5.6s on 1.73GHz Pentium-M and 1GB RAM configuration, respectively. This is caused by the fact that many computation efforts are needed for the recovery of shape parameters. The 3D face recognition rate which uses shape + texture data must be higher than that which uses only texture data. But, the experimental result shows that considering the shape coefficients doesn't improve the face recognition performance meaningfully.

Consider a more general context in which many users are not required in practical application such as access control in a home or office. In the second test, a total of 20 trials were done by randomly selecting 21, which is equal to the subject number of the web camera based 2D face database, out of 62 probe images. These results are reported in **Fig. 14** and **Table 4**. The number in the parenthesis indicates the correct hit count out of 21. In cases of 'texture' and 'both', the accuracy rate averages 92.1% and 93.3%, respectively. Also, the 'Both' case showed the same or more accurate performance against the 'Texture' case in most trials except for two trials named #15 and #19. This result will compare with those of the Combination A-2 in the afterward.

Lastly, we examine how the number of shape and texture basis has influences on the system performance. The test is achieved in a manner that one out of the shapes and textures is forced to be expressed with a fixed basis number for variations on basis number of the other. As shown in **Table 5**, the basis of shape and texture was proportionally selected to 50% and 70% of the whole, respectively. **Table 5** shows the results on the computation time of the fitter. When only the basis number of texture is increased, the fitting time is in a very small range increased as well. However, in the reverse case, it was found that the variation of the fitting

time is by far larger over the former. This is also caused by the fact that most computation time in model fitting is taken to the recovery of the shape parameters as mentioned in the first test. Therefore, we can conclude that the basis number of the shape is the factor having a very important impact on the computational efficiency of the system.

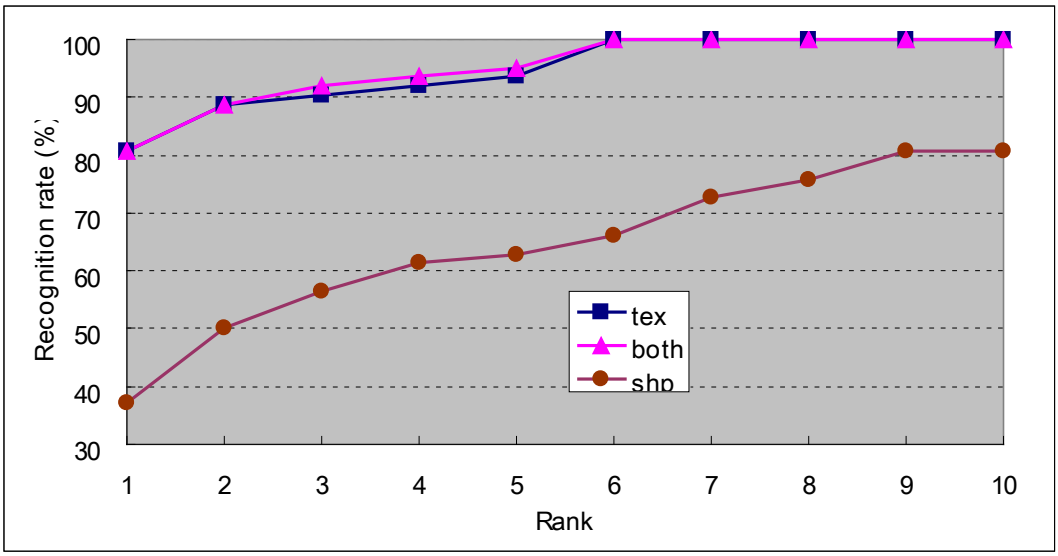

Fig. 14. CMC to rank 10 in A-1 with 50% shape and texture basis.

Table 4. The result on in A-1.

Test	Accuracy Rate (hit number)		Test	Accuracy Rate (hit number)	
	Texture	Both		Texture	Both
#1	85.7(18)	90.5(19)	#11	90.5(19)	90.5(19)
#2	90.5(19)	90.5(19)	#12	100.0(21)	100.0(21)
#3	85.7(18)	90.5(19)	#13	90.5(19)	95.2(20)
#4	95.2(20)	95.2(20)	#14	85.7(18)	85.7(18)
#5	100.0(21)	100.0(21)	#15	95.2(20)	90.5(19)
#6	100.0(21)	100.0(21)	#16	85.7(18)	95.2(20)
#7	90.5(19)	90.5(19)	#17	85.7(18)	90.5(19)
#8	100.0(21)	100.0(21)	#18	100.0(21)	100.0(21)
#9	90.5(19)	90.5(19)	#19	90.5(19)	85.7(18)
#10	85.7(18)	90.5(19)	#20	95.2(20)	95.2(20)
Average				**92.1(19.35)**	**93.3(19.6)**

Table 5. The computation time of fitting with respect to variations of shape and texture.

Texture Basis Selection Rate (%)	Shape Basis Selection Rate (%)	Recognition Rate (%)	Total Fitting Time (sec)	Average Fitting Time (sec)

10	50	81	321	5.18
30	50	91	343	5.53
50	50	93	348	5.61
70	50	100	356	5.74
90	50	100	377	6.08
70	10	38	55	0.89
70	30	57	155	2.50
70	50	63	353	5.69
70	70	73	668	10.77
70	90	81	1048	16.90

5. Conclusion and Future work

In this paper, we presented an efficient 3D face representation algorithm using the pixel-to-vertex map (PVM) as a face recognition algorithm. On the basis of the PVM, 3D face data including 30,000 ~ 40,000 vertices could be efficiently represented with 4,822 vertices. We have generated a 3D morphable model using a hundred 3D face image databases (each 3D face image is synthesized with 4,822 vertices using PVM). Then, shape and texture coefficients of the 3D face model were estimated by fitting into an input face using the Inverse Compositional Image Alignment (ICIA) algorithm.

We have collected the 3D face database from a stereo-camera based device for 3D model generation and performance evaluation. Based on the experimental results, the proposed face representation and recognition algorithm presents a reasonable recognition rate while maintaining computation complexity. Our proposed algorithm shows better recognition performance with real-time implementation while Romdhani et al. [4] presented 30 seconds fitting time with less recognition accuracy. Our future work will include optimization of texture and shape information to enhance recognition accuracy.

Acknowledgement

This research was conducted with the support of the Seoul R&BD Program (10581).

References

[1] T. Papatheodorou and D. Rueckert, “Evaluation of 3D face recognition using registration and PCA,” *Lecture Notes in Computer Science*, vol. 3546/2005, pp. 997-1009, 2005. Article (CrossRef Link).

[2] V. Blanz and T. Vetter, “A morphable model for the synthesis of 3D faces,” in *Proc. of Computer Graphics, Annual Conference Series (SIGGRAPH)*, pp. 187-194, 1999. Article (CrossRef Link).

[3] V. Blanz and T. Vetter, “Face recognition based on fitting a 3D morphable model,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1063-1074, 2003. Article (CrossRef Link).

[4] S. Romdhani and T. Vetter, “Efficient, robust and accurate fitting of a 3D morphable model,” in *Proc. of IEEE International Conference on Computer Vision*, pp. 56-66, 2003. Article (CrossRef Link).

[5] S. Baker and I. Matthews, "Equivalence and efficiency of Image alignment algorithms," in *Proc. of CVPR*, pp. I-1090 - I-1097, 2001. Article (CrossRef Link).

[6] T. Sim, S. Baker and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1615-1618, 2003. Article (CrossRef Link).

[7] P. Phillips, H. Moon and P. Rauss, "The FERET evaluation methodology for face recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no.10, pp. 1090-1104, 2000. Article (CrossRef Link).

[8] P.J. Phillips, P. Grother, R.J. Micheals, D.M. Blackburn, E. Tabassi and J.M. Bone, "FRVT 2002: evaluation report," *Mar*. 2003. http://www.frvt.org/FRVT2002/documents.htm

[9] S. Shan, W. Gao, B. Cao and D. Zhao, "Illumination normalization for robust face recognition against varying lighting conditions," in *Proc. of IEEE Workshop on AMFG*, pp. 157-164, 2003. Article (CrossRef Link).

[10] X. Xie and K.-M. Lam, "Face recognition under varying illumination based on a 2D face shape model," *Pattern Recognition*, vol. 38, no. 2, pp. 221-230, 2005. Article (CrossRef Link).

[11] Kwang Ho An and Myung Jin Chung, "Pose-robust face recognition based on texture mapping," in *Proc. of the 17th IEEE International Symposium on Robot and Human Interactive Communication*, pp. 65-70, 2008. Article (CrossRef Link).

[12] R. Jain, R. Kasturi and B. Schunck, "Machine Vision", McGraw-Hill, 1995.

[13] B. Horn and B. Schunk, "Determining optical flow," Artificial Intelligence, vol. 17, no. 1-3, pp. 185-203, 1981. Article Article (CrossRef Link).

[14] B. K. P. Horn, H. M. Hilden and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *Journal of the Optical Society of America A*, vol. 5, no. 7, pp. 1127-1135, 1988. Article (CrossRef Link).

[15] X. Lu, A. Jain and D. Colbry. "Matching 2.5D face scans to 3D models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 31-43, 2006. Article (CrossRef Link).

[16] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuroscience*, vol. 3, no. 1, pp.71-86, 1991. Article (CrossRef Link).

[17] T. Vetter and T. Poggio, "Linear Object Classes and Image Synthesis from a Single Example Image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 733-742, 1997. Article (CrossRef Link).

[18] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces versus fisherfaces: recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711-720, Jul. 1997. Article (CrossRef Link).

[19] B. Bascle and A. Blake, "Separability of pose and expression in facial tracking and animation," in *Proc. of Sixth International Conference on Computer Vision*, pp. 323-328. 1998. Article (CrossRef Link).

[20] S. Romdhani, N. Canterakis and T. Vetter, "Selective vs. global recovery of rigid and non-rigid motion," *Technical report, CS Dept., Univ. of Basel*, 2003. Article (CrossRef Link).

[21] J. Kittler, A. Hilton, M. Hamouz and J. Illingworth, "3D assisted face recognition: A survey of 3D imaging, modeling and recognition approachest," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp.20-26, 2005. Article (CrossRef Link).

[22] A. S. Georghiades, P. N. Belhumeur and D. J. Kriegman, "From few to many: illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, June 2001. Article (CrossRef Link).

[23] C. Liu, J. Yuen, A. Torralba, J. Sivic and W. T. Freeman, "SIFT Flow: dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PP, no. 99, pp. 1-1, *2010*. Article (CrossRef Link).

[24] http://www.wikipedia.org/

[25] H. Moon and P. Phillips, "Computational and performance aspects of projection-based Face Recognition Algorithms," *Perception*, vol. 30, no. 3, pp. 303-321, Mar. 2001. Article (CrossRef Link).

[26] L. Zou, S. Cheng, Z. Xiong, M. Lu and K.R. Castleman, "3D face recognition based on warped example faces," *IEEE Trans. Information Forensics and Security*, vol. 2, no. 3, pp. 513-529, Sept.2007. Article (CrossRef Link).

Kanghun Jeong received a B.S. degree from Sejong University, Seoul, in 2006, an M.S. in 2008, and is currently continuing a Ph.D. degree in Computer Engineering from Sejong University. His research interests include computer vision, pattern recognition, multimodal biometrics, information security and face detection/recognition based on a smartphone environment.

Hyeonjoon Moon received a B.S. degree in Electronics and Computer Engineering from Korea University, Seoul, in 1990, an M.S., and Ph.D. degrees in Electrical and Computer Engineering from the State University of New York at Buffalo, in 1992 and 1999, respectively. From 1993 to 1994, he was a systems engineer at Samsung Data Systems in Seoul, Korea. From 1996 to 1999, he was a research associate at the US Army Research Laboratory in Adelphi, Maryland. From 1999 to 2002, he was a senior research scientist at Viisage Technology in Littleton, Massachusetts. Currently, he is an associate professor of Computer Engineering at Sejong University in Seoul, Korea. His research interests include computer vision, pattern recognition and biometrics.