# Individualized 3D Face Model Reconstruction using Two Orthogonal Face Images

SunHee Weon, *Member, IAENG*, SungIl Joo, and HyungIl Choi

***Abstract*— In this paper, there are two main topics to generate a 3D facial model from 2D face images. The first topic is to extract facial components such as eyebrows, eyes, nose, ear, mouth and face contour on the face images, and the second is to create 3D individualized face model using extracted facial components information at previous step. The generated final face model has an individualized shape, realistic skin texture and deformable structure for facial animation. In this paper, we propose novel approach to detect facial components automatically using a active contour extraction algorithm(Active Contour Model, Snake) for facial information. And how to generate adaptive individualized 3D face model from detected facial components are explained and result of 3D face models are discussed.**

***Index Terms*—3D face modeling, Active contour model (snake), Facial feature extraction, Template snake, 3D reconstruction.**

## I. Introduction

THE 3D face modeling technique treated in this paper, is applied in a wider range such as virtual conference or plastic surgery, while of which need is getting increased, but it has limit to get a realistic facial model at ease and simply. Currently there are 2 general ways to get 3D face model. One is to obtain 3D image of human head from specific equipments as 3D scanner. Since the data scanned from virtual reality is managed, animation-structure combination is essential for this. However, this animation-process is not proper for unskillful users to make a virtual 3D face model, since it requires additional data which is not directly offered in scanning process. The other study is about re-forming faces with corresponding pixels using stereo images or video sequences[6][7], while another study is underway to generate 3D face model with front and flank photo images[1,10-13]. However, it is very difficult to make 3D face model by 2 face photos of which video pixels do not have any 3D data. x and y coordinates are from front image while y and z are from flank image, but 3-dimensional coordinate-values of forehead and cheek with no specific points do not correspond with feature points of 2-dimensional photo images, while eyes, nose, mouth and chin line are possible. Meanwhile, facial features offer lots of specific data about people, making 3D surface of it. Therefore, extracting features from photo images and meeting two images are main subjects of studies to make realistic face models. Currently manual, semi-automatic, and automatic ways to extract 2D features from photo image are under study. Although semi-automatic way strongly works to make 3D model, but it requires users to understand meaning of each feature, which is used to manually adjust location of features. A study about automatic process is vital hence, for a system to make face model that is more simply available. However, an automatic process to powerfully generate face models given no matter how images has not been realized yet, which is resulted from limits of various skin colors, races, and ages. Consequently, this study is to extract facial features more accurately and promptly, and to make more natural 3D features from 2D ones.

The automatic 3D individualized face modeling system is described in this paper consists of two modules: Facial components extraction and 3D face modeling. First, facial component extraction is described in section 2. How to deform the 3D generic face model and generate texture map are explained in section 3. Also, Experimental results are given in section 4. Section 5 concludes the paper.

## II. Extract to Facial Components

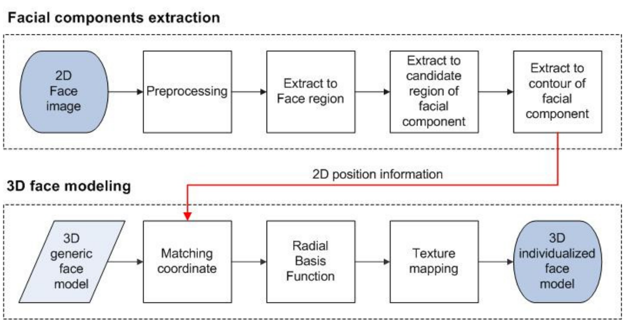



Fig. 1. System architecture of proposed 3D face modeling

Our 3D individualized face modeling system consists of two steps as shown [Figure 1]: 1) facial components extraction and 2) 3D face modeling. We use color space for skin-color filtering processing and then analysis about facial feature by extracted facial components. Obtained feature of facial components used for generate 3D individualized face model. In following section, we explained that method for detect face region and extract to contour of facial components.

Manuscript received July 19, 2012; revised Aug. 10, 2012. This work was supported by the Seoul R&BD Program(SS110013).

SunHee Weon, Dr., is a Lecturer in the Department of Global Media, Soongsil University, Seoul, Republic of Korea(e-mail : nifty12@ssu.ac.kr).
SungIl Joo, Mr., is with the Department of Global Media, Soongsil University, Seoul, Republic of Korea(e-mail : sijoo82@ssu.ac.kr).
HyungIl Choi, Dr., is a Professor in the Department of Global Media, Soongsil University, Seoul, Republic of Korea(e-mail : hic@ssu.ac.kr).

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

### A. *Segmentation of facial components*

The most widely used way to detect facial area is using skin color data. Skin color data is the most effective since it is promptly dealt with and not much influenced by geometric change of face. The optimal YCbCr color model was adopted in this paper after many experiments, and facial area is extracted by modeling Cb, Cr values only which are directly linked with skin color, since Y component, meaning brightness, is much influenced by light at the time of extracting facial area.[9] As a result of sampling data by YCbCr color model, the range of skin color was determined like equation (1). Prior to classify skin color area with this reference map, facial contour is firstly drawn from input facial image through snake, an algorithm to extract dynamic contour. Within facial area drawn from snake, facial area and facial components are decided by skin color. It is facial area when Cb, Cr values of pixel (x, y) are in the suggested range of equation (1), and it is facial components if not, so facial components area can be segmented by determining it as 0(black color).

[Figure 2] separately detected facial area and facial components using skin-conference map made by equation (1). (a) is potential facial area which is skin color, while (b) and (c) detected potential facial components within facial area with MER type using equation (2), by projecting it horizontally and vertically.

$$\text{skin region}\begin{cases} 77 < cb < 117 \\ 143 < cr < 173 \end{cases} \tag{1}$$

$$H_i = \sum_{j=0}^{Y}\sum_{i=0}^{X}\frac{x(j,i)}{255}, H_i = \sum_{i=0}^{X}\sum_{j=0}^{Y}\frac{y(i,j)}{255} \tag{2}$$

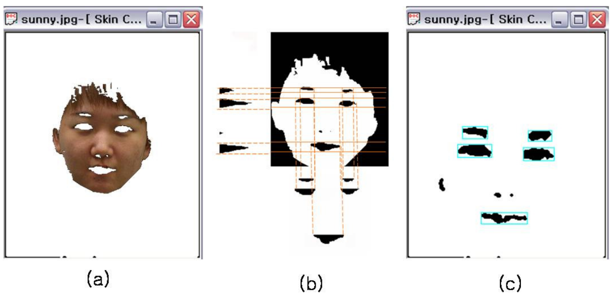



Fig. 2. Detected candidate area of face and face components
(a) Classified face and face components region (b) Result of vertical and horizontal projection
(c) Candidate area of face components

### B. *Extract to contour for facial components*

Contour of each face-specific component is detected within pre-drawn potential facial components area using snake which defines initial template model, and edge should be regarded incomplete considering each local feature of facial components. This paper resolved it by giving it a change to energy term of current snake algorithm, and suggested to $E_{eye\_snake}$ to detect eyes and eyebrows contour and $E_{mouth\_snake}$ to detect mouth contour like equation (3), (4), to separately define the last energy item $E_{image}$ according to facial components. Especially it is hard to draw accurate edge of eyes and eyebrows if they are wrinkled or covered by hair, so the image item is defined to extract edge by combining the first and second edge operators. Constants were determined as 1.0, 1.7, 0.5, and 0.5, after experiments. Image item of mouth edge was defined only by the first edge detector since it is clearer than eyes or eyebrows. Constants for equation (4) are defined as 1.0, 1.7, and 1.0, after experiments.

$$E_{eye\_snake} = \int_0^1 \left( \alpha(s)\left|\frac{dv}{ds}\right|^2 + \beta(s)\left|\frac{d^2v}{ds^2}\right|^2 - \left(r_1\left|\nabla I(v) + r_2\left|\nabla^2 I(v)\right|\right|\right) \right) ds \tag{3}$$

$$E_{mouth\_snake} = \int_0^1 \left( \alpha(s)\left|\frac{dv}{ds}\right|^2 + \beta(s)\left|\frac{d^2v}{ds^2}\right|^2 - \left(r_1\left|\nabla I(v)\right|\right) \right) ds \tag{4}$$

## III. 3D INDIVIDUALIZED FACE MODEL

In this research, we have used 3Ds Max, to create the generic face model just like in [Figure 3]. This face figure is made of 926 vertexes in total, and 1797 polygons. By analyzing the file from this face model through exporting into ASE file which is one of the export format, we were able to find the coordination of each vertex and based on this, we have created the generic model.

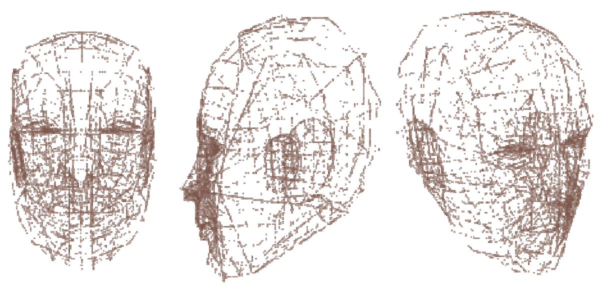


Fig. 3. 3D Generic face model

### A. *Deformations of generic face model*

For the modeling of 3D face, we need the process of matching the generic model into the face of individuals. In this paper, we are going to call the points shown on the wire mesh, the control points, to be able to control the certain parts of the pictorial image of the face.

TABLE I
AN FACIAL FEATURE CLASSIFIED BY FACIAL PORTION OF CONTROL POINT

| Part / View | Contour | Eyebrows | Eye | Nose | Mouth | Total |
|---|---|---|---|---|---|---|
| Front | 12 | 4 | 12 | 4 | 7 | 39 |
| Side | 6 | 2 | 3 | 3 | 4 | 18 |

The formation of head, and face is very complicated and detailed, so we have made the control points to match into each of the facial components of face in parts, considering the relations of elements according to its shape, size, and facial feature components, and as shown in table 1, it can be divided into 6 detailed parts, such as eyes, eyebrows, nose, ears, lips,

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

head, and face contour. Also, by using control point, we made it possible for the accurate control of matching face models, not affecting other groups and maintaining its individuality.

Radial Basis Function is recommended for smooth model surface, in the process to transform the shape of 3D generic face model based on data about extracted 2D facial features. The number of points from corresponding two models must be the same in general transforming process, but better function is expected in this way only if accurate locations of corresponding points are found out although they are not the same[14][15]. Since facial features drawn from previous clause contains correct locations and shapes of facial features, the shape of model can be transformed through matching vertex with 3D generic face model mash. RBF function is used to transform generic face model. 39 front and 12 flank facial feature points are used as it is pre-defined, and they are clustered according to influences on each feature points. RBF function is an interpolation where each feature point move to correspond with those moved to template snake, while vertex influenced by each feature point move to the middle area. If f(x’) is a feature point to transform, RBF interpolation function is defined as F(x’,c’). x' is a vector designating feature point, and c' is weight being influenced by each feature point. RBF interpolation function also can by expressed by equation (5), where N is number of feature points, h(r) is basis function of RBF, and r means euclidean distance between feature and vertex.

$$F(\vec{x}) = \sum_{i=1}^{N} c_i h(r) \tag{5}$$

Gaussian function is used in this paper as basis function of RBF function, of which equation is (6).

$$F(\vec{x}) = \sum_{i=0}^{N} c_i e^{-\left(\frac{r_i^2}{\sigma_i}\right)} \tag{6}$$

Parameters of weight and length are required to perform equation (6), and weight $c_i$ along with length parameter $\sigma_i$ must be figured out since feature points are pre-determined $(x_i, y_i, z_i)$. When $x^{source}$ is initial location of feature points and $x^{target}$ is transported one, weight $c_i$ is figured out by equation (7), since displacement of all feature points are noticed.

Length parameter $\sigma_i$, which is related with the range influencing on vertex of each feature-point cluster, is figured out like equation (8), using the longest among euclidean distances and weight of each cluster. Therefore, bigger weight on vertex near feature points and smaller weight on those far from feature points are added, to smoothly transform vertex as feature points move.

$$c = h^{-1} x^{target} \tag{7}$$

$$\sigma_i = \max \left\| x_i^{source} - x_i \right\| \times weight \tag{8}$$

### *B. Texture map Generation*

A texture about appearing plane is available by suitable personal face model, by controlling template snake-applied feature points. In rendering, the final stage of facial modeling, shading or texture mapping is used. Shading is hard to express delicate part like faces, since lighting or features figure out pixel values. However texture mapping is to lengthen or shorten model surface on 3D coordinate system, which can almost realistically express the actual model. This paper therefore, forms view-independent texture map by composing 2D images like following. View-independent texture map can promptly proceed with rendering despite changed view, and can get accurate-colored 3D face model that is independent of lighting.

Image located between 3D coordinates of face model mash and the texture space of 2D facial image, is defined by cylinder mapping. After figuring (u,v) by reflecting 3D coordinate of facial model mash on cylinder, followed by $(x_i, y_i)$ about j camera where fixed point p on the mash is reflected on 2D plane image, then $I_i(x_i, y_i)$, color value for p, is figured out. Final color combination of texture map about each pixel (u,v), is as equation(10).

$$I(u,v) = \sum_i W_i(u,v) I_i(u,v) \tag{10}$$

Here, $W_i(u,v)$ is weight according to contribution to the image input by j camera, and composed weight of mash about each facial image is fixed based on internal dimension of camera-view unit vector and perpendicular unit vector of 3D mash. In case corresponding points (u,v) are out of image, weight is fixed as 0, while total weight is regulated as 1. This paper had made a texture map only by 2 pieces of image, not by 3 pieces for current experiments. In other words, texture data of left face image is reflected on corresponding point to the standard of front image coordinate, while empty space of back-side texture data is filled with combining texture data of feature points, which allows more prompt rendering process.

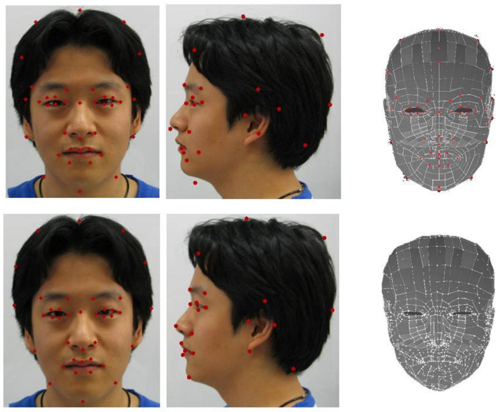
## IV. EXPERIMENTAL RESULT

In this paper MFC library and OpenCV ver 1.0, an image-process open library based on Visual C++ 6.0, were realized and experimented in the system where OpenGL accelerating board is installed on Pentium Ⅳ 3.4GHz CPU. [Figure 4] is a result of edge-finding for face contour and feature components out of overall image by template snake, of which repetition was within 10 times to correctly find this edge and location [Figure 5].

In [Figure 4] (c) also displays transformed result of standard face model into personal one, using extracted feature

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

points. In this process feature points and control points in 3D model move along the edge, while personal face model is generated. So iteration to extract location of feature points are same with that of control points.



(a) (b)

Fig. 4. Individualized face model (a) extract facial feature points (b) deformation face model

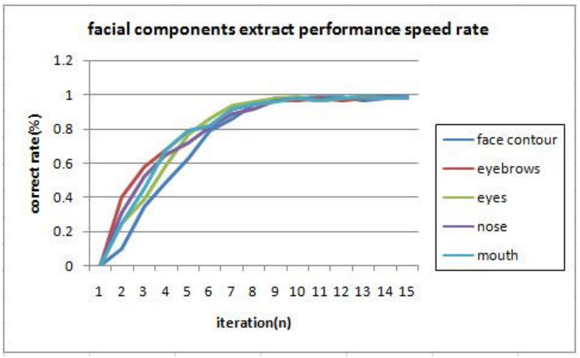



Fig. 5. Facial components extract process time for correct rate

[Figure 6] is comparison of three or more directions and two directions face images, where is no big difference between them in being natural. Table 2 is comparison the speed of manual extraction of facial features with 3D face model made by AAM, and we see that 2D image automatically extract facial features skipping texture corresponding process of right and back side images, allowing speedier results. [Figure 7] is comparison of result from the algorithm proposed in this paper with from that in other papers. (a) has been able to make a face model similar with real image, by forming shapes from front and flank sides and by texture-mapping in advance. In case of (b) front and flank shapes are not separately determined in other algorithm processes but shape and texture on flank side were corresponded as it is to make face model, which leads to distinctively different face model.

(a) (b)

Fig. 6. Compared with method using multi-view image (a) 2-directions (b) 3 or more directions

(a) (b)

Fig. 7. Compared with other method[16] (a) Our method (b) other method[16]

## V. CONCLUSION

In this paper 2 facial images crossing at right angle were used to automatically make 3D face model, and texture map was obtained from 2 actual images to present how to generate realistic personal face model. Different from current way to extract facial components by handwork which needs learning, template snake algorithm has hugely reduced editing time of 3D face model, which can dynamically extract various facial components.

Meanwhile, texture map obtained from 2 pieces of image made generation realistic face model possible. The way to make texture map with current 2 pieces of image can produce much natural and realistic personal face model, since it generated it only after corresponding the shapes of face model in advance, different from obtaining results by making symmetry regardless of different individual faces. To reflect more realistic facial image of texture, a study about color distribution technique along adjacent edge in color revision process for 2 neighboring images, would be required.

TABLE II
COMPARISON PROCESS TIME WITH OTHER METHOD

| 3D Modeling / Process | Manually Face Modeling | Automatically Face Modeling (AAM) | Proposed Method |
|---|---|---|---|
| Facial feature extraction | off the chart | 15.3 sec | 10.1 sec |
| Face model matching | 89.0 sec | 19.0 sec | 10.1 sec |
| Texture map generation | off the chart | 22.1 sec | 12.3 sec |
| Rendering | off the chart | 0.01 sec less than | 0.01 sec less than |
| Total process time | 89+$\alpha$ sec | 56.31 sec | 31.21 sec |

## REFERENCES

[1] T. Akimoto, Y. Suenaga, and S.W. Richard, "Automatic creation of 3Dfacial models", IEEE Computer Graphics and Applications, IEEE Computer Society Press, Silver Spring, MD, 1993.
[2] T. Russ, C. Boehnen, and T. Peters, "3D Face Recognition Using 3D Alignment for PCA", IEEE Conf. on Computer Vision and Pattern Recognition, Vol.2, pp. 1391-1398, 2006.
[3] A-Nasser Ansari, and Mohamed Abdel-Mottaleb, "Automatic facial feature extraction and 3D face modeling using two orthogonal view with application to 3D face recognition", The Journal of the Pattern Recognition, Vol. 38, pp. 2549-2563, 2005.
[4] Z. Mandun, M. Linna, X. Y. Zeng, and Y. S. Wang, "Image-Based 3D Face Modeling", Proc. of Int'l Conf. on Computer Graphics, Imaging and Visualization, pp. 165-168, 2007.
[5] A. Ansari and M. Abdel-Mottaleb, "3-D Face Modeling Using Two Views and a Generic Face Model with Application to 3-D Face Recognition", Proc. of IEEE Conf. on Advanced Video and Signal Based Surveillance (AVSS'03), pp. 37, July 2003.

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[6] P. Fua, and Y.G. Leclerc, "Taking advantage of image-based and geometry-based constraints to recover 3-D surfaces", Comput. Vision Image Understanding 64(1), pp. 111–127, July 1996.
[7] P. Fua, "Regularized bundle-adjustment to model heads from image sequences without calibration data", Int. J. Comput. Vision 38(2), pp. 153–171, July 2000.
[8] M. Kass, A. Witkin, and D. Terzopoulus, "Snakes : Active Contour Models", International Journal of Computer Vision, 1(4), pp. 321-331, 1987.
[9] Sunhee Weon, KeunSoo Lee, and Gyeyoung Kim, "Contour Extraction of Facial Feature Components Using Template Based Snake Algorithm", ICCSA 2007, LNCS 4705, Part I, pp. 1034-1044, 2007.
[10] H.H.S. Ip, and L. Yin, "Constructing a 3D individualized head model from two orthogonal views", Visual Comput. 12, pp. 254–266, 1996.
[11] T. Kurihara, and K. Arai, "A transformation method for modeling and animation of the human face from photographs", in: Proceedings of Computer Animation’91, Tokyo, pp. 45–58, 1991.
[12] W. Lee, P. Kalra, and N. Magnenat-Thalmann, "Model based face reconstruction for animation", in : Proceedings of MMM’97, World Scientific Press, Singapore, pp. 323–338, 1997.
[13] W. Lee, and N. Magnenat-Thalmann, "Fast head modeling for animation", J. Image Vision Comput. 18 (4), 355–364, 2000.
[14] Jun-young Noh, Douglas Fidaleo, and Ulrich Neumann, "Animated Deformation with Radial Basis Functions", in Proc. the ACM Symposium on Virtual Reality Software and Technology (VRST’00), New York, NY, USA: ACM Press, pp. 166–174, 2000.
[15] Chen Chen, and Edmond Cyril Prakash, "Personalized Cyber Face: A Novel Facial Modeling Approach using Multi-Level Radial Basis Function", Proc. of the 2005 Int'l Conf. on Cyberworlds (CW’05), pp. 475-482, Nov. 2005.
[16] Don-Hee Kim, Jong-Hyun Yoon, and Jong-Seoung Park, “Generating Face Textures using Real Images”, KIISE conference 2007, Vol.34, No.2(B), pp. 156-161, 2007.
[17] Sunhee Weon, and Gyeyoung Kim, “A Study on Generating Texture Map using Two Views of 2D Face Images and Automatic 3D Face Modeling”, KIISE conference 2008, Vol.35, No.2(A), pp. 231-232, 2008.

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)