A System of Robust Real-time Face Tracking and Modeling from Video

Ronghua Liang[1),2)], Chun Chen[1)], Zhigeng Pan[1)] , and Jiajun Bu[1)]

1) College of Computer Science, Zhejiang University , Hangzhou,310027, P.R.China
rhliang2001@hotmail.com
2) Institute of VR and Multimedia, Hangzhou Inst. of Electronics Engineering , Hangzhou,310037, P.R.China

Abstract

Real-time face tracking and modeling from video have been a challenge in computer graphics and computer vision for many years. And it is widely used in the field of virtual reality such as Human-Machine interaction, computer games and film, virtual showman, etc. The two issues of real-time face tracking and modeling should be taken into consideration: how to generate facial features in the first frame of video automatically and how to improve the algorithmic efficiency of real-time face tracking and modeling. We develop a system to track face and model face from video in real time. Our system is integrated with auto-generation of face features of the first frame, image pyramid algorithm with Kalman filter and 3D reconstruction and face pose modeling. Experimental results show the high prospect of this algorithm.

Key words face tracking, Kalman filter, image pyramid algorithm, 3D reconstruction.

1. Introduction

Real-time face tracking and modeling from video have been a challenge in computer graphics and computer vision. And it is widely used in the field of Human-Machine interaction, computer games and film, virtual showman, etc. The issues of real-time face tracking and modeling are twofold:

1) Auto-generation of facial features in the first frame. Now prevalent approach to generate facial features is to create image corners. To obtain the face features is very difficult and imaccurate if the face anatomy is not considered

2) The algorithmic efficiency of face tracking and modeling. As we know, most of methods can not meet the requirement of real-time human-machine interaction.

Many researchers use motion capture to track and model face and body to generate face animation and body animation [1-4]. Firstly, sensors are used in motion capture to track 3D motion of body or face. Then, after data processing, skeleton or face creation, the animated character is generated by mapping data onto skeleton or face. But we have to create 3D models of individuals manually.

The approach of tracking and modeling face in real time by software technique includes three stages. The fist stage is to acquire the face features of the first frame, the second stage is to get vision disparity by SfM, and the third stage is to reconstruct the 3D face model.

Thomas S. Huang et al [5] proposed an explanation-based algorithm of facial motion tracking based on a piecewise Bézier volume deformation model (PBVD). With this model, basic facial movements, or action units, are interactively defined. By changing the magnitudes of these action units, animated facial images are generated. The limitation is to align the features of the first frame and the 3D model manually. They also presented an algorithm to track natural hands movements based on sequential Monte Carlo by integrating hand motin constraints [6]. The algorithm is easy to extend to other articulated motion capturing tasks.

Yu Zhang et al [7] proposed a physical-based approach based on anatomical knowledge for real-time facial expression animation. The facial model incorporates a physical-based approximation to facial skin and a set of anatomically-motivated facial muscles. The skin is modeled by a mass-spring system with nonlinear springs which have biphaisic stress-strain relationship to simulate the elastic dynamics of real facial skin. Facial muscles are modeled as forces deformation the spring mesh based on the Aus(Action Unites) of the Facial Action Coding System(FACS). Their approach can not reconstruct the expression from videos in real-time.

J. Ström et al [8] developed a real-time system for tracking and modeling of face by using an analysis-by-synthesis approach. Feature points of the first frame in the face-texture are selected based on image Hessians. The features are tracked using normalized correlation. The result is fed into an extended Kalman filter to recover camera geometry, head pose, and structure from motion. One main difference between our work and theirs is that we used general 3D model to fit the first frame based on anatomical knowledge and that we used Kalman filter to estimate the 2D image motion.

In this paper, an algorithm is presented to track face by integrating image pyramid algorithm and Kalman filter. First, facial features of first frame are acquired by Plessey corner detector based on anatomical knowledge and general 3D model of human face. And then corner disparity is obtained by image pyramid algorithm and Kalman filter, finally 3D model is reconstructed by SfM.

The rest of this paper is organized as follows: In section 2 the detailed description and analysis of the algorithm are given, the experimental results are shown in Section 3 and the related conclusions are given in Section 4.

2. Algorithm Description

2.1 Algorithm Overview

The fully integrated system is shown in Figure 1. After face detection, 3D general model alignment and image corners selection, face features of the first frame can be generated automatically. The three process make full use of the anatomical knowledge of human face and image property. To improve the algorithmic efficiency of face tracking, Kalman filter is employed to estimate the motion of the head, and it also resolves the problem of occlusion during face tracking. Finally, individual model is fit into the video. According to 3D reconstruction, we can get the rigid transformation (rotation and translation), then the motion 3D model can be obtained.

2.2 Face Alignment of the First Frame

1) Face detection

When detaching the head from the original images includes, two problems should be taken into consideration: how to detach the whole figure from the background and how to get the figure’s face excluding its body. First, the images in RGB color system are converted into HSV color system. The figure can be distinguished from the background according to Hue histogram in HSV color system. Though Hue different skin colors are with different hue., the statistics of their Hue is alike. Face is separated from other parts of figure according to the property of face, such as color of face, the distance between eyes and nose tip, etc.

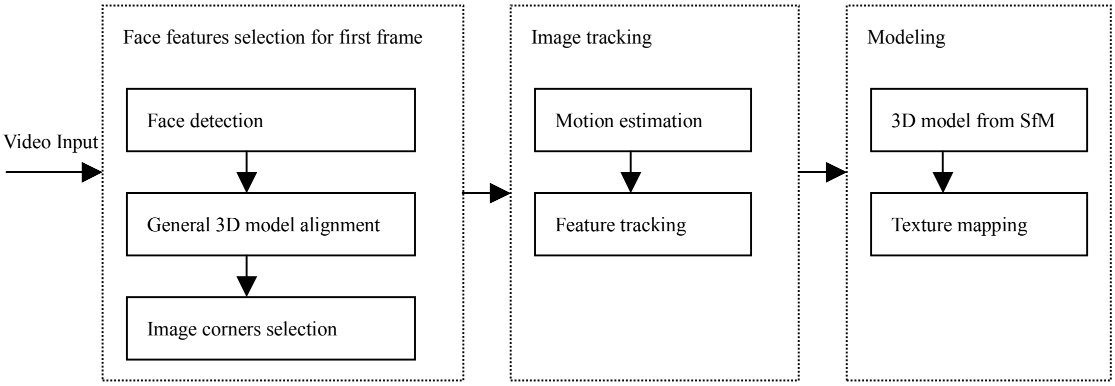


Fig. 1: Algorithm overview.

2) Features selection by general 3D model fitting first frame

The 3D individual face model can be denoted as F ={*S*, *C*, *L*, *T*}, where S is 3D vertex coordinates of general face model and it is denoted as *S* = { *V1,V2,…Vn*}, *Vi=(x,y,z)*, *C*=(*c1,c2,…ck*) is the collection of face features index, *L* is the deformation parameters for individual face, and *T* is the factors of rigid transformation which is rotation matrix and translation vector . Face features selection by general 3D model fitting in the first frame consists of two stages: The first stage is to project the general 3D model into 2D image, and the second stage is to deform the whole meshes of 3D general model.

Let l_x be the facial width of individual, and let l_y be the facial height of of individual. $O(O_x,O_y,O_z)$ L_x,L_y,L_z are center point, facial width, facial height, facial depth of 3D general model, respectively. For every vertex of general model, such as (V_x,V_y,V_z), the new position in 2D image by projection is $V_i^{'}=(V_i-O_i)*l_i/L_i \qquad where\ i=x,y.$

3) Corners detection

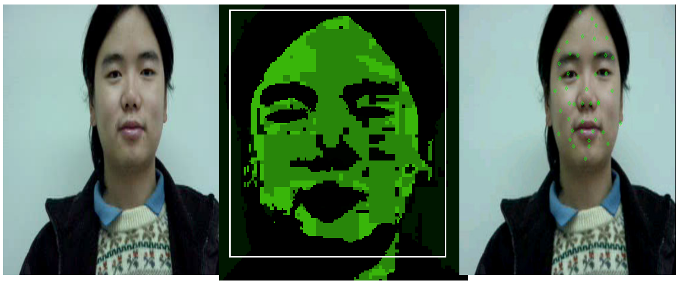
The Plessey corners detection [9] is employed in our algorithm. We can define the matrix:

$$G=\begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \qquad (1)$$

where I_x,I_y are the deviation of pixel in X and Y direction in images, respectively. Corners function is defined $R=\det G-k(traceG)^2$. According to Harris proposal, *k* is equal to 0.04. The sorted list with the 68 highest *R* is selected as the candidates of face features. According to results of 3D general model fitting, face features can be obtained by resolving the following equation:

$$Min\sum_{k}\sum_{j=1,2}(V_k^j-U_k^j)^2 \qquad (2)$$

where V_k^j represents the coordinate of 3D general model projection in U and V direction, respectively, and V_k^j represents the coordinate by corners detection in U and V direction respectively. By displacement of corners, face features can be obtained. Experimental results of face alignment of first frame are shown in Figure 2.



(a)first frame (b)face detection (c)features selection, features is denoted by green points.

Fig. 2: Face alignment of first frame

2.3 Face Features Tracking

1) Motion estimation with Kalman filter

Kalman filter is used to estimate the motion of human face, and it constrains image search space, so the tracking efficiency is greatly improved. In this paper, standard Kalman filter is employed to tracking face features in 2D images. Face motion sequence is assumed as a dynamic system, and the acceleration of features is assumed constant. The velocity can be obtained by Kalman filter time update and estimation update. Without loss of generality, Feature **p**=(*u*,*v*)^T can be denoted by: **p**=**p**'+ η, where **p**' is the real position, **p** is the measurement position, and η represents measurement noise, and it is assumed to be white noise with *zero-mean* (statistically)

Guass probability distributions and covariance matrix is

$$\Lambda_\eta = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix}.$$

The state vector has six dimensions $\mathbf{s} = \begin{bmatrix} u(k), v(k) \\ \dot{u}(k), \dot{v}(k) \\ \ddot{u}(k), \ddot{v}(k) \end{bmatrix}$, where *u(k)* and *v(k)* are the image coordinate of features in U and V direction of image respectively, $\dot{u}(k)$ and $\dot{v}(k)$ are the velocity in U and V direction respectively, $\ddot{u}(k)$ and $\ddot{v}(k)$ are the acceleration in U and V direction respectively. The state equation can be defined as

$$s(k+1) = \mathbf{F}s(k) + \Gamma\,\mathbf{n}(k) \qquad (3)$$

and measurement equation can be defined as

$$Z(k)=\mathbf{H}s(k) + \eta\,(k) \qquad (4)$$

Where

$$F = \begin{bmatrix} 1 & 0 & \Delta t & 0 & \frac{1}{2}\Delta t^2 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & \frac{1}{2}\Delta t^2 \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \Gamma = \begin{bmatrix} \frac{1}{2}\Delta t^2 & 0 \\ 0 & \frac{1}{2}\Delta t^2 \\ \Delta t & 0 \\ 0 & \Delta t \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$ and K=0,1,2,…, represents the sequence number, △*t* is the time interval, and △*t*=t$_{k+1}$-t$_k$, in our system, △*t*=1/24s. n(k) is white noise with *zero-mean* (statistically), Guass probability distributions white noise with *zero-mean* (statistically)

$$E[n(k)] = 0 \; and \; E[n(k)n(j)] = Q, \; for \; i \neq j$$

Gain equation [10] is represented as

$$K = \Lambda_\eta H^T [H\Lambda_\eta H + Q]^{-1} \qquad (5)$$

Kalman filter is implemented by "time update" and "measurement update".

2) Tracking face based on image pyramid algorithm

Kalman filter is an estimate algorithm. The measurement and tracking can be succeeded by image pyramid algorithm. Let **u(x,y)** be a pixel in an image I(x,y), and its neighbor field is denoted by $\delta_{\mathbf{u}}(\mathbf{v},\mathbf{w_x},\mathbf{w_y}) = \{\mathbf{v}(\mathbf{p},\mathbf{q}) | \; \|\mathbf{p}-\mathbf{x}\| \le w_{\mathbf{x}}, \|\mathbf{p}-\mathbf{y}\| \le \mathbf{w_y}\}$.

The gradient matrix of the neighbor field of the pixel $\mathbf{r}(\mathbf{p_x},\mathbf{p_y})$ is

$$G_{\mathbf{r},\delta} = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \qquad (6)$$

where I_x, I_y is the derivative in X and Y direction, respectively.

Let I(x,y) and J(x,y) be the two corresponding images , the algorithm aims to search for a pixel **u** in I(x,y) and its corresponding pixel **v** in J(x,y). we denote disparity vector $\mathbf{d} = \mathbf{v} - \mathbf{u}$, which is the same for its neighbor field $\delta_{\mathbf{u}}(\mathbf{v},\mathbf{w_x},\mathbf{w_y})$. The aim of corner matching is to search for *d* which minimizes:

$$\varepsilon(d) = \varepsilon(d_x,d_y) = \sum_{x=u_x-\omega_x}^{u_x+\omega_x} \sum_{y=u_y-\omega_y}^{u_y+\omega_y} (I(x,y) - J(x+d_x, y+d_y))^2 \quad (7)$$

We define the pyramid representation of a generic image $I^0, I^1 ......I^L$. Let I(x,y) be the "zeroth " level image, then Equation 7 can be modified as:

$$\varepsilon^L(d^L) = \varepsilon^L(d_x^L,d_y^L) = \sum_{x=u_x^L-\omega_x}^{u_x^L+\omega_x} \sum_{y=u_y^L-\omega_y}^{u_y^L+\omega_y} (I^L(x,y) - J^L(x+g_x^L+d_x^L, y+g_y^L+d_y^L))^2 \qquad (8)$$

where $g^L = \left[g_x^L, g_y^L\right]^T$ and $d^L = \left[d_x^L, d_y^L\right]^T$ are the Lth pyramid optical flow and motion vector, and their relation is denoted by $g^{L-1} = 2(g^L + d^L)$.

The detailed algorithm is described in Ref. 11

2.4 3D Model Reconstruction

1) Generation of 3D coordinate of face features

Define (x,y) as the normalization coordinate for pixel $m(u,v)$ in I(x,y) if

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = A^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \qquad (9)$$

where A is the intrinsic matrix of camera.

Let (x,y) and (x',y') be the normalization coordinates of corresponding matches, then epipolar constraint is :

$$(x',y',1)E(x,y,1)^T = 0, \qquad (10)$$

where

$$E = [\mathbf{t}]_\times \mathbf{R} \qquad (11)$$

and $[\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$. (**R,t**) is the 3D rigid transformation(rotation **R** and translation **t**).

We presume the motion of camera is rigid and translation is occurred after rotation from origin, then the motion we get is unique. We resolve the problem of structure from motion based on Zhang's work [12] as follows:

Step 1: Estimate the essential parameters with 8-point algorithm (see Equation 9 and Equation 10).

Step 2: Compute **R**, t according to Equation 11.

Step 3: Refine the parameters of E and **R** and **t** by minimizing the sum of squared distances between points and their epipolar lines.

Step 4: Reconstruct the corresponding 3D points for corner points in image by using camera calibration.

2) Creation of 3D individual model

According to the results in the previous section, let u^i represent transformation of feature point p^i . The individual model can be obtained by deformation of general 3D model and interpolation of the features. Then resolve the following equation:

$$f(p) = \sum_i c^i \Phi(\left\| p - p^i \right\|) + Mp + t \qquad (12)$$

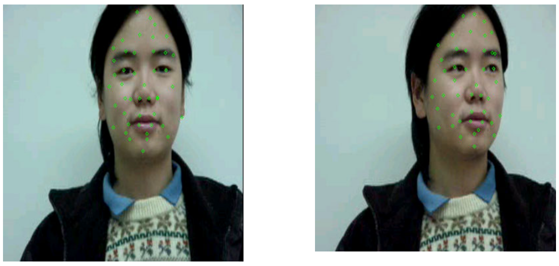
Where $\Phi(r) = e^{\frac{-r}{64}}$, M and t represent 3 x 3 projective matrix and 3 x 1 vector respectively. And calculate coefficient c^i and M and t. A linear system (see Equation 12) submitted to equation (see Equation 14) also have to be resolved:

$$\begin{cases} u^i = f(p^i) \\ \sum_i c^i = 0 \\ \sum_i c^i \cdot p^{i^T} = 0 \end{cases} \qquad (13)$$

$$\begin{bmatrix} 1 & 1 & \cdots & 1 & 0 & 0 & 0 & 0 \\ p_x^0 & p_x^1 & \cdots & p_x^{N-1} & 0 & 0 & 0 & 0 \\ p_y^0 & p_y^1 & \cdots & p_y^{N-1} & 0 & 0 & 0 & 0 \\ p_z^0 & p_z^1 & \cdots & p_z^{N-1} & 0 & 0 & 0 & 0 \\ \Phi_0^0 & \Phi_1^0 & \cdots & \Phi_{N-1}^0 & p_x^0 & p_y^0 & p_z^0 & 1 \\ \Phi_0^1 & \Phi_1^1 & \cdots & \Phi_{N-1}^1 & p_x^1 & p_y^1 & p_z^1 & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \Phi_0^{N-1} & \Phi_1^{N-1} & \cdots & \Phi_{N-1}^{N-1} & p_x^{N-1} & p_y^{N-1} & p_z^{N-1} & 1 \end{bmatrix} \begin{bmatrix} c_x^0 & c_y^0 & c_z^0 \\ c_x^1 & c_y^1 & c_z^1 \\ \vdots & \vdots & \vdots \\ c_x^{N-1} & c_y^{N-1} & c_z^{N-1} \\ M_0^0 & M_1^0 & M_2^0 \\ M_0^1 & M_1^1 & M_2^1 \\ M_0^2 & M_1^2 & M_2^2 \\ t_x & t_y & t_z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ u_x^0 & u_y^0 & u_z^0 \\ u_x^1 & u_y^1 & u_z^1 \\ \vdots & \vdots & \vdots \\ u_x^{N-1} & u_y^{N-1} & u_z^{N-1} \end{bmatrix} \qquad (14)$$

The 3D individual model is assured to have minimal energy value and the efficiency is improved after linear system is resolved.

Experimental results are shown in Figure 3.



(a) first frame (b) 75th frame

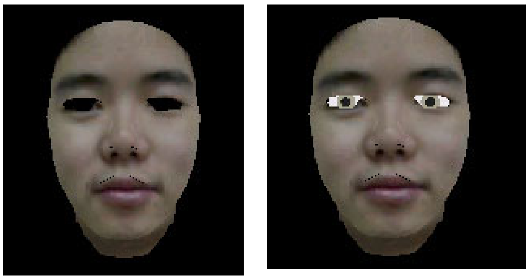
Fig. 3: Results of features tracking

2.5 Eyes modeling

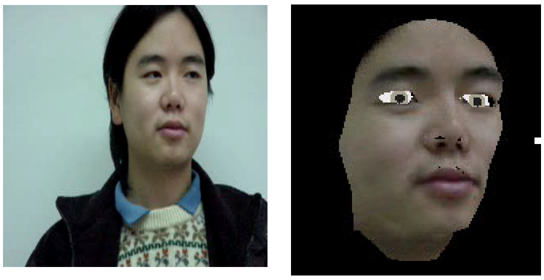
Eyes modeling is to generate realistic-looking 3D face model. Each 3D model of two eyes is represented as two cycles and triangles surrounding the cycles. Eye model is denoted by *EM*, and *EM={R1,R2,Tr}*, where *R1*=r1^2, R2=r2^2, and r1<r2. *R1* and *R2* consist of 12 triangles respectively. *Tr* consists of 36 triangles surrounding *R2* and connects eye socket.

3 Experimental Result

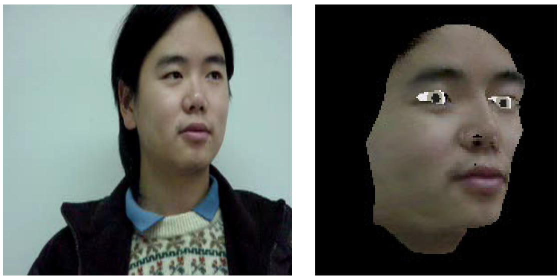
The experimental system is implemented with Visual C++ 6.0 under Windows 2000, and the images used in this experiment are obtained by a camera. 3D individual model transformation (relative to the first frame) can be generated by the method presented in Section 2.4. Experimental results of reconstruction for Some frames are show in Figure 4, with eyes modeling, the realistic-looking face model can be obtained.



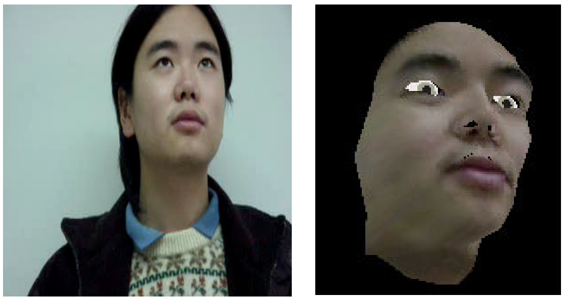
(a) Result of comparison of 3D reconstruction for the first frame, the left picture is 3D model without eyes modeling, and the right picture with eyes modeling.



(b) Result of 3D reconstruction of the 44th frame.



(c) Result of 3D reconstruction of the 75th frame.



(d) Result of 3D reconstruction of the 94th frame.

Fig. 4: Results of real-time reconstruction

4. Conclusions

In this paper, an algorithm is presented to track face by integrating image pyramid algorithm and Kalman filter. First, facial features of first frame are acquired by Plessey corner detector based on anatomical knowledge and general 3D model of human face. And then corner disparity is obtained by image pyramid algorithm and Kalman filter, finally 3D model is reconstructed by SfM. Experimental results show the high prospect of this algorithm. The future research work includes mouth model, hair model, eyes model, and generation of realistic model.

Acknowledgments

This work was partly supported by NSFC grants 60083009 and 60203013, excellent young NSF of Zhejiang province grants 40008, and Zhejiang Provincial Natural Science Foundation of China grants 601043, 602134.

Reference

1 C. Hang, I. Lin, M. Ouhyong: “High Resoluation Calibration of Motion Capture Data for Realistic Facial Animation,” *Journal of software*, *China*, Vol. 11, pp.1141- 1150(2000).

2 J. Lee, J. Chai, P. S. A. Reitsma: “Interactive Control of Avatars Animated With Human Motion Data,” *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, pp.491-500(2002).

3 Y. Li, T. Wang, H.Y. Shum: “Motion Texture: A Two-Level Statistical Model for Character Motion Synthesis,” *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH,* pp.465-472(2002).

4 L. Kovar, M. Gleicher, F. Pighin: “Motion Graphs,” *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH,* pp.473-482(2002).

5 H. Tao, T. S. Huang: “Explanation-based Facial Motion Tracking Using a Piecewise Bézier Volume Deformation Model,” *IEEE CVPR* (1999).

6 Y. Wu, J. Y.Lin, T. S. Huang: “Capturing Natural Hand Articulation,” *IEEE ICCV* (2001).

7 Y. Zhang, E. C. Prakash, E. Sung: “Real-time Physically-based Expression Animation Using Mass-Spring System,” *IEEE Computer Graphics Internationa*, Hong Kong (2001).

8 J. Ström, T. Jebara, S. Basu. A. Pentland: “Real Time Tracking and Modeling of Faces: An EKF-based Analysis by Synthesis,”*IEEE ICCV*(1999).

9 C. Harris and M. Stephens: “A combined corner and edge detector,”. *Proceeding 4th Alvey Vision Conference*, 189~192(1988).

10 R. E.Kalman: “A New Approach to Linear Filtering and Prediction Problems,”. *Transaction of the ASME—Journal of Basic Engineering*, Vol.82 pp.35-45(1960).

11 Ronghua Liang, Chun Chen, Zhigeng Pan and Hui Zhang: “A new algorithm for 3D facial model reconstruction and its application in VR,”. *Proceedings of the International Conference on VR and its Application in Industry (VRAI*), pp.119-124(2002).

12 Z.Zhang: “A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry,” *Artificial Intelligence*, Vol.78, pp.87-119(1995).