3D Matching applied ICP Algorithm for 3D Facial Avatar Modeling Using Stereo Camera

Hyo-seok Kang, Sung-hoon Yu, and Mignon Park*

ICS Laboratory(B723), The Department of Electrical and Electronic Engineering,
Yonsei University, 134, Shinchon-Dong, Sudaemoon-Gu, Seoul, 120-749, Korean
khs3538@yonsei.ac.kr, mignpark@yonsei.ac.kr

Abstract. This paper is proposed by 3D facial avatar modeling using stereo camera. We use the adaboost algorithm for facial detection to get 3D facial data. We make the 3D facial modeling to match extractive 3D data with justified 3D facial avatar. We use ICP algorithm for matching 3D data and control the number of vertex and feature point for efficiency and accuracy. Feature point is selected within outline points using canny edge method. We propose to make personal 3D facial avatar for reducing expense and time.

Keywords: 3D facial avatar, Stereo camera, Vertex, Iterative Closest Point(ICP).

1 Introduction

Animation market has been getting bigger these days since the users want more reality in 3D animation. 3D avatar is used in the cyber space as a guidance for shopping and searching. The use of 3D avatar is increasing dramatically in a time of passing over from personal computer period to personal robot period because of developing humanoid robot. UI(User Interface) will be used to control robot easily and comfortably.

The growth of today's 3D avatar market shows that it's great potential. To create the Key Frame Animation discontinuously in 3D animation or 3D game, it requires much time and effort. Therefore, 3D modeling has been developed along with many other tools[1] [2]. It requires a long time to make the animation because current tools can only allow drawing motions one by one. This results the low rate of using avatar in the animation industry. Engineers need more capabilities in strong technology to improve the animation industry. This paper is shown 3D avatar modeling using stereo camera to create each Key Frame Animation. It creates intimacy and uses UI easily.

This work is organized as follows. In section 2, related works from the extraction of 3D data using stereo camera are discussed. Section 3 explains 3D Matching in justified model and extractive 3D data using ICP algorithm. Experimental results and discussions are presented in Section 4. Finally, conclusion is given in Section 5.

* Corresponding author.

Y. Lee et al. (Eds.): ICOST 2010, LNCS 6159, pp. 222–225, 2010.

2 3D Data Extraction Using Stereo Camera

Stereo camera can get 3D vertexes using disparity between two cameras. We obtain depth information to control disparity. It is available data in distance from 30cm and 3m for detecting a face[3]. In this paper, we need 3D facial vertex data. So we use to detect face using the Adaboost algorithm offered OpenCV. 3D data is provided (X,Y,Z) value of each pixel. we use Canny Edge extraction method for robustness in environment because extractive outline includes feature points. It is shown in Fig 1.

(*a*) (*b*) (*c*)

Fig. 1. Detection of face using Adaboost algorithm(*a*), 3D image in stereo camera(*b*), and feature vertexes using canny edge extractive method(*c*)

3 3D Matching Using ICP

We need 3D vertexes matching for merging extractive data with justified 3D facial data. In this paper, We use Iterative Closest Point(ICP) algorithm that is usually used to match 3D vertexes.

In summation, ICP Algorithm is as following:

1. Get the closest vertex set corresponded to each data Point in model.
2. Calculate the Registration between obtained vertex sets.
3. Apply a conversion for matching vertex sets from process 2.
4. Repeat from process 1 if Registration Error overflows an acceptable error range.

In process 1, we use an Euclidean distance when we calculate a distance.

$$\mathrm{d(P,M)} = \|\mathrm{P}-\mathrm{M}\| = \sqrt{(\mathrm{x_p}-\mathrm{x_m})^2 + (\mathrm{y_{p,}}-\mathrm{y_m})^2 + (\mathrm{z_p}-\mathrm{z_m})^2}, \qquad (1)$$

where $\mathrm{P} = (\mathrm{x_p}, \mathrm{y_{p,}}, \mathrm{z_p})$, $\mathrm{M} = (\mathrm{x_m}, \mathrm{y_{m,}}, \mathrm{z_m})$.

We obtain the closest vertex set of model corresponded to each data point that is calculated for a distance from an image. In process 2, we get rotation matrix and translation matrix based on process 1. First of all, we define the number of data point(P) as $\mathrm{N_p}$ and model point(M) as $\mathrm{N_m}$. The purpose of process 2 is to get minimization of $\mathrm{q_R}, \mathrm{q_T}$ in equation 2.

$$\mathrm{f(q)} = \frac{1}{\mathrm{N_p}} \sum_{\mathrm{i=1}}^{\mathrm{N_p}} \|\mathrm{m_i} - \mathrm{R(q_R)p_i} - \mathrm{q_T}\|^2 \qquad (2)$$

The center of mess of data points and model points is represented by equation 3.

$$\mu_p = \frac{1}{N_p}\sum_{i=1}^{N_p} p_i,\ \mu_m = \frac{1}{N_m}\sum_{i=1}^{N_m} m_i \tag{3}$$

The cross-covariance matrix $\sum pm$ of data points and model points is equation 4.

$$\sum pm = \frac{1}{N_p}\sum_{i=1}^{N_p}\left[(p_i - \mu_p)(m_i - \mu_m)^t\right] = \frac{1}{N_p}\sum_{i=1}^{N_p}[p_i {m_i}^t] - \mu_p {\mu_m}^t \tag{4}$$

We define $A_{ij} = (\sum pm - \sum pm^t)_{ij}$ and $\Delta = [A_{23}\ A_{31}\ A_{12}]^T$ to determine equation (5).

$$\left(\sum pm\right) = \begin{bmatrix} tr\left(\sum pm\right) & \Delta^T \\ \Delta & \sum pm - \sum pm^t - tr\left(\sum pm\right) I_3 \end{bmatrix}, \tag{5}$$

where I_3 is 3×3 Identity matrix.

From equation (5), we can get q_R that is corresponding unit eigentvetor to the largest eigenvalue of matrix $Q(\sum pm)$.

$$q_R = [q_0\ \ q_1\ \ q_2\ \ q_3] \tag{6}$$

And then, q_T is defined in equation (7).

$$q_T = \mu_m - R(q_R)\mu_p, \tag{7}$$

where

$$R(q_R) = \begin{bmatrix} {q_0}^2 + {q_1}^2 - {q_2}^2 - {q_3}^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & {q_0}^2 + {q_2}^2 - {q_1}^2 - {q_3}^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_1 q_2 - q_0 q_3) & {q_0}^2 + {q_3}^2 - {q_{21}}^2 - {q_2}^2 \end{bmatrix}$$

In process 3, we convert data using q_T and q_R in process 2. In other words,

$$P_{k+1} = q_k(P_0) \tag{8}$$

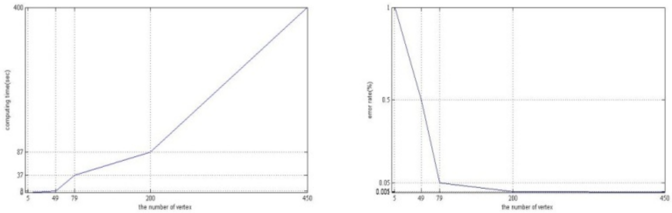
We define threshold τ in process 4. And then we repeat from process 1 to 3 until satisfying $d_k - d_{k-1} < \tau$. We can know how many rotation and translation in these processes. Rotation matrix(R) and translation matrix(T) are defined as equation 9.

$$T = q_T = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}, \qquad R = R(q_R) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{23} & r_{33} \end{bmatrix} \tag{9}$$

4 Experimental Result and Consideration

In this paper, we extracted facial 3D point using stereo camera. Our purpose is that we make personal 3D facial avatar to match extractive data with some justified models. Detecting user's face using stereo camera and extracting 3D data are represented in Fig 4.

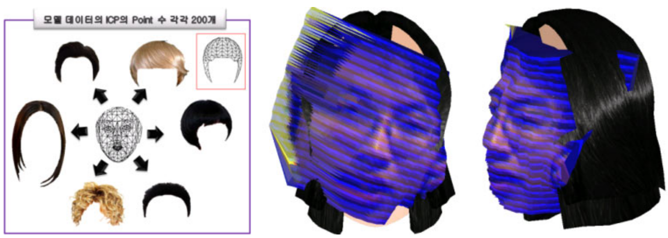
For matching extractive data from stereo camera with justified model, we used ICP algorithm in this experiment. We studied the number of vertex for applying ICP algorithm in real-time. Fig 2. is a result for error rate and computing time.



(*a*) (*b*)

Fig. 2. Computing time(*a*) and accuracy(*b*) for the number of vertex

We determined 200 feature vertexes and matched 3D facial data with justified model in Fig 3.



(*a*) (*b*) (*c*)

Fig. 3. Justified models(*a*) and matching 3D facial avatar in front of face(*b*)and side(*c*)

5 Conclusion

The object of this paper implements personal 3D face avatar. The modeling method using the stereo camera makes that the cost can be reduced and 3D avatar can be rapidly produced. However, implementation of the whole face is impossible. So we suggested the method to make completed 3D facial avatar, which was matched extractive data using stereo camera with modeled avatar. The number of a vertex and the abstraction of characteristic points are important for the matching about computing time and accuracy. The feature point and the number of a vertex were found with trial and error. The 200 vertexes have the best result at an accuracy and speed. The performing speed is about 87 seconds, and an error came out less than 0.01% without falling into Local minimum.

References

1. Noh, J.: A survey of Facial modeling and Animation Techniques, USC Technical Report, pp. 99-105 (1998)
2. Wen, Z., Huang, T.S.: 3D Face Processing: Modeling, Analysis and Synthesis Series. The International Series in Video Computing 8, 10–160 (2004)
3. Pollard, S., Mayhew, J., Frisby, J.: PMF: A stereo correspondence algorithm using a disparity gradient limit. Perception 14, 449–470 (1985)