Journal of Korea Multimedia Society Vol. 12, No. 6, JUNE 2009(pp. 804-814)

Online Face Avatar Motion Control based on Face Tracking

Li Wei[†], Eung-Joo Lee[††]

ABSTRACT

In this paper, a novel system for avatar motion controlling by tracking face is presented. The system is composed of three main parts: firstly, LCS (Local Cluster Searching) method based face feature detection algorithm, secondly, HMM based feature points recognition algorithm, and finally, avatar controlling and animation generation algorithm. In LCS method, face region can be divided into many small piece regions in horizontal and vertical direction. Then the method will judge each cross point that if it is an object point, edge point or the background point. The HMM method will distinguish the mouth, eyes, nose etc. from these feature points. Based on the detected facial feature points, the 3D avatar is controlled by two ways: avatar orientation and animation, the avatar orientation controlling information can be acquired by analyzing facial geometric information; avatar animation can be generated from the face feature points smoothly. And finally for evaluating performance of the developed system, we implement the system on Window XP OS, the results show that the system can have an excellent performance.

Key words: Face Recognition, Face Detection, Avatar Animation, HMM

1. INTRODUCTION

Creating and controlling a realistic face avatar is as of today still a great challenge as our computer is deficient to simulate the complicated of facial behaviors and we are all expert to distinguish even subtly unnatural facial expressions. Especially, for both gaming and collaborative work applications, such as the Second Life or 3D Poker, to create and control face avatar could be more challenging. Face avatar embodied in those online games should be controlled through using user's face behavior, such as: expression, movement or rotation [1], method by using face expression to control avatar mainly utilize the feature points to describe the face features, and extract face expression by the confirmed feature points, thus the face expression can be described by the relationship of these feature points and through this way, the avatar animation can be created and controlled, as discussed by Xiaozhou Wei etc. [2]. However until now, after a decade's effort, robust and realistic real time face tracking and generation still faced a big challenge. The difficulty lies in a number of issues including the real time face feature tracking under a variety of imaging conditions (e.g., lighting variation, pose change, self-occlusion and multiple non-rigid features deformation), and the real time realistic face modeling and animation using a very limited number of feature parameters. And also on the other ways, traditional method modeled the head motion as a 3D rigid motion with the local skin deformation [3,4], the linear motion tracking method cannot represent the rapid head motion and dramatic expression change accurately. This is not a suitable method for dealing with real time face video for controlling avatar motion which needs quick response and accurately performance. The appearance-driven approach requires a significant number of training data to enu-

※ Corresponding Author : Eung-Joo Lee, Address : (608-711) Department of Information & Communications Engineering, Tongmyong University, Busan, Korea, TEL : +82-51-629-1143, FAX : +82-51-629-1129, E-mail : ejlee@tu.ac.kr

Receipt date : Dec. 30, 2008, Approval date : June 18, 2009

[†] Department of Information Communication Engineering, TongMyong University, Korea
(E-mail : li_wei1983@naver.com)

[††] Department of Information Communication Engineering, TongMyong University, Korea

merate all the possible appearances of features. The model based approach [5,6] assumes the knowledge of a specific object is available, meanwhile the requirement of frontal facial views and constant illumination limited its application. All above tracking methods have shown certain limitations for accurate face feature tracking under complex imaging conditions.

So in this paper, In order to solve these problems, we present a novel approach which uses face geometric information to estimate the head movement and rotation information, and use mesh structure to represent the facial feature, the face expression feature will be extracted by tracking the feature points. And for detecting the feature points, we will present a novel method which we called LCS (Local Cluster Searching). The method detects feature points through analyzing a small local region of face. In each local region, we need only to judge that if it is an object region, background region or edge region. The algorithm is just concerned with a small region, and each region is independent to the others, so it cannot be influenced by the noise in the image. When each feature point is classified, we can select the edge points and object feature points, then use HMM based recognition algorithm to classify them for getting the face features: mouth, eyes, nose etc. Avatar orientation controlling information can be acquired by calculating rotation angles of head which is based on facial geometric information. And avatar animation can be synthesized by tracking the facial feature points. Each animation origins from the facial feature points, in the animation approach, the no-feature points will be created dynamically and considered as a new feature point for creating next no-feature point. And the new no-feature point position is confirmed by calculate the weight of its' pro-feature points. Through this way, the avatar expression motion can be represented smoothly and closely to real face expression. The face avatar motion control system is outlined in Figure 1,

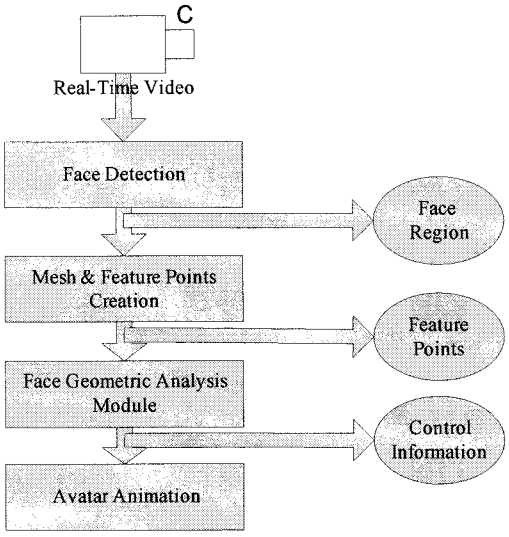


Fig. 1. Framework of the Face Avatar Motion Control System.

which consists of five modules: (1) Face Detection Module; (2) Mesh & Feature Points Creation Module; (3) LCS Detection Module; (4) Face Geometric Analysis Module and (5) Avatar Animation Module. And finally we will evaluate the efficiency of our proposed system through analyzing the experiment result and give the conclusion.

2. HAAR-LIKE FEATURES BASED FACE DETECTION

Because the performance of the system is usually influenced by the reliability of face detection, so a surpassing algorithm selected for face detection is important. Usually face detection can be performed based on several cues: skin color (for color images or videos which include faces), motion (videos include faces), facial shape, facial appearance, or a combination of these cues. The most successful face detection algorithm is based on the facial appearance. The appearance-based face detection algorithm scanned at all the possible locations and scaled by a sub-window, and then face detection is done by classifying the pattern in the

sub-window as either face or non-face. The face/no-face classifier is learned from face and non-face training examples using statistical learning method. We use AdaBoost-based face detection method [7] since AdaBoost is the most successful learning method in terms of detection accuracy and processing speed. AdaBoost is one of the boosting methods. The boosting algorithm generates different weak classifiers by changing example weights and constructs a strong classifier by combining weak classifiers.

AdaBoost-based face detection uses Haar-like features as patterns, Haar-like feature is one of the digital image feature used in object recognition. This feature set considers rectangular regions of the image and make integral value of the pixels in the region. The result is used for categorizing whether the sub-window includes a face or not. Since the binary rectangular filters are applied by changing aspect ratio, size and location inside the sub-window of size 24 x 24, the total number of filters are tens of thousands. An effective rectangular filter for face detection is only a small part of them. Hence, the AdaBoost algorithm is used to select effective filters. Examples of selected binary rectangular filters are shown in Figure 2.

We generate the classifiers (*weak classifiers*) based on features obtained from selected filters, where the weak classifier determines whether the sub-window includes a face or not. In the learning process of weak classifiers, we use a number of face and non-face images to perform effective learning. The final classifier (*strong classifier*) is generated by combining weak classifiers with weights of their reliability.

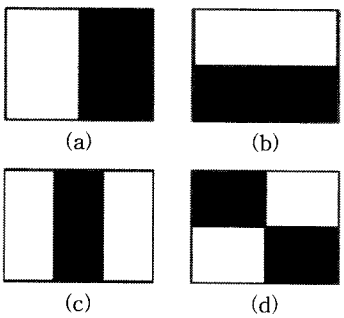


Fig. 2. Basic Haar-like Feature for Detecting Face: (a). Two-rectangle Features in Horizontal Direction; (b). Two-rectangle Features in Vertical Direction; (c). Three-rectangle Feature and (d) Four-rectangle Feature.

Usually a good classifier spends a lot of time to give result because there are many features are considered. The input patterns, however, include both simply (simple background) and difficulty (complex background) recognized patterns. For recognizing the simple patterns, we just need some simple feature in order to examine all features. But with a general problem, the classifier must check all features, which is extracted from the learning process, even the recognizing pattern belong to the simple group. So, it is a time-consuming process. Cascade of classifiers is constructed to solve this problem. The cascade of classifiers is a series of classifiers are applied to every sub-window. Cascade tree includes many of stage (layer), each stage is a stage classifier. Each stage classifier is trained by the negative samples which previous stage classifier recognized incorrect. Figure 3 shows an example of the cascade connection of k-strong classifiers. The performance of face detection is increased since the sub-window is recognized as a face only all the strong classifier as a face. This strategy can significantly reduce the computation time of face detection since most of robustness of sub-windows are reject by the first strong classifier.

A face in an image may be detected several

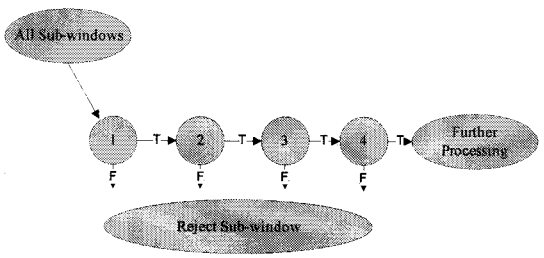
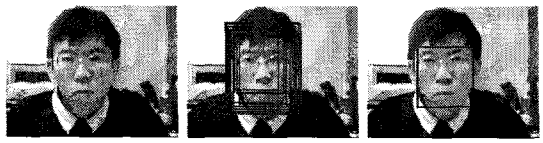


Fig. 3. A Cascade of k-strong Classifiers



(a) (b) (c)

Fig. 4. Example of Face Detection: (a) Original Image, (b) Detected Face Regions, and (c) Combined Face Region.

times at neighbor locations or on multiple scales as shown in Figure 4 (b). Non-face regions may also be detected even if the cascade connection of strong classifiers is used. Addressing these problems, we use the fact that the correct regions are not. After detecting face regions, we select the correct regions and combine them to detect a correct face region. Figure 4 illustrates an example of face detection using the AdaBoost algorithm. The face region is automatically detected from the image by using the above algorithm.

3. MESH GENERATION AND LCS METHOD BASED FACIAL FEATURE DETECTION

For a face candidate region, the facial features (mouth, nose, eyes) will need to be detected. We can know that, if we assume the face region is a background, each feature will be an independent and connective object region in this background. So for detecting these features, the only thing we should to do is to detect the object from the background which is based on face region. Also as we can observe that, appearance of each object (feature) concentrates to a center of the object region. Therefore we can define several feature points in the face region, and use it to judge that if an appointed small region of it is an object region, edge region or background region. Usually if the appointed region is background, it can have a similar illumination value with the background and value of the region changed not frequently. So the region can have a lesser variance result. Meanwhile, if an appointed region has a big difference value with background and variance of the region is small, we can confirm the feature point is an object feature point. Because value of edge region changed frequently, so its' variance value is usually greater than both the object region and background.

For getting the feature points, we divide the face region rectangle in both horizontal and vertical direction. Then connect the corresponding points in both directions, the cross points can be considered as the feature points. As shown in Figure 5.

We assume $P_i(x, y)$ is a feature point inside the face region, R_i is the appointed region which used $P_i(x, y)$ as the center point, W'_i and H'_i is width and height of each sub-region, the region size should not be greater than the divided face region ($W'_I \leq [W/M]$, $H'_I \leq [H/N]$, where W, H is width and height of face region; M, N is total division number in both horizontal and vertical direction). In order to judge that the edge region we apply the following equation to each appointed region.

$$\sigma_i = \sqrt{s_i^2} = \sqrt{\frac{1}{(W'_i * N'_i) - 1} \sum_{j=1}^{W'_i * N'_i} (x_{ij} - \bar{x}_i)^2} \quad (1)$$

$$\sigma_{Th} = \frac{1}{M * N} \sum_{i \leq M * N} \sigma_i \quad (2)$$

Here, x_{ij} denotes the gray pixel value in the appointed region, and $\bar{x}_i$ is mean value of the region. σ_i denotes the variance value and σ_{Th} is threshold

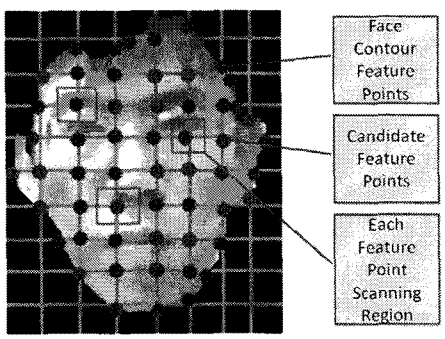
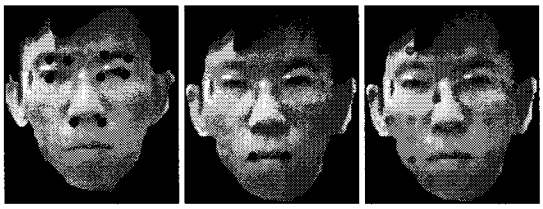


Fig. 5. Facial Feature Points Determination Diagram (M=10, N=8 in the Face Region)



(a) (b) (c)

Fig. 6. Feature Points Classification Result: (a). Edge Feature Points; (b). Object Feature Points and (c). Background Feature Points

value used for classifying edge and no-edge region.

If σ_i is much greater than σ_{Th}, we can judge it as an edge region, otherwise it is an object or the background region. Feature point of this region can be considered according to the region result.

The no-edge regions have two kinds: object region and background region, to separate them, we can compare pixel difference between the appointed region and the face region average value. As following equation shows that:

$$\overline{X} = \frac{1}{W * H} \sum_{i \le W * H} X_i \tag{3}$$

$$\omega_i = \left| \overline{x_i} - \overline{X} \right| \tag{4}$$

We can use a threshold value T to judge the result ω_i, if ω_i is greater than T, the region can be judged to an object region, otherwise it is the background region.

The background feature points can be ignored, and facial feature can be represented by the edge points and object features points. Thus we can classify the mouth, eyes, and nose according to face geometric lay-out.

4. FACE TRACKING BASED ON FACIAL FEATURE POINTS GEOMETRY

In this section, we will use the detected feature points to analyze the face feature geometric information, the geometric information can be utilize to confirm that if a person is laughing, angry or smiling, the avatar animation can also been generated according to the facial geometric information.

As we know, the mouth is necessary for face tracking while it doesn't reflect the tester's expression all the time. Sometime, eye, and eyelid will play together to represent a completed expression. So in our study, the tracked features are divided into two groups experientially. The facial motion feature of the lips and jaw are classified into visual speech set while the others are processed as expression one. At the frontal part, we have detected the facial feature points exactly. Then each facial feature points set can be vectorization and finally, the vector set can be tracked using the trained model.

4.1 Facial Feature Vectorization

For a set of n tracked feature points $p_1,...p_n$ on the face define the face geometric information, assuming the neutral face is $\overrightarrow{f_0}$, for one frame, the transition of feature points can be represented as the vector $\vec{f} = \overrightarrow{f'} - \overrightarrow{f_0} = [f_1^x, f_1^y, ..., f_n^x, f_n^y]^T$, $\overrightarrow{f'}$ is the coordination for current frame. Then as we have explained that, the mouth feature points can be grouped into visual speech set and other face feature points can be grouped into expression set respectively. Correspondingly, $\overrightarrow{m^s} = [p_1^s, ..., p_\lambda^s]^T$ and $\overrightarrow{m^e} = [p_{\lambda+1}^e, ..., p_N^s]^T$. In our case, N is number of total face feature points and λ is number of mouth feature points.

With the frames from 1 to c, we can represent the visual information as the facial motion configuration described above. The input vectors are stacked for the different frames to form two stream vector M^S and M^E where to be used as the input of HMM.

$$M^S = [\vec{m}_1^{\,s}, \vec{m}_2^{\,s}, ..., \vec{m}_c^{\,s}] \tag{5}$$

$$M^E = [\vec{m}_1^{\,e}, \vec{m}_2^{\,e}, ..., \vec{m}_c^{\,e}] \tag{6}$$

4.2 HMM based on Facial Feature Recognition

A discrete-time Hidden Markov Model λ can be

viewed as a Markov model whose states cannot be explicitly observed: Each state has an associated probability distribution function, modeling the probability of emitting symbols from that state. More formally, a HMM is defined by the following entities:

- S = $\{S_1,S_2,\cdots,S_N\}$ a finite set of hidden states;
- The transition matrix A=$\{a_{ij}, 1\leqq j\leqq N\}$ representing the probability of going from state S_i to state S_j,

$$a_{ij} = P[q_{t+1} = S_j \mid q_t = S_i] \; 1 \le i, j \le N \tag{7}$$

With $a_{ij}\geqq 0$, $\sum_{j=1}^{N} a_{ij} = 1$;

- The emission parameters B =$\{b(o|S_j)\}$, indicating the probability of emission of the symbol o when the system state is S_j. In this paper we employ continuous HMMs: $b(o|S_j)$ is represented by a Gaussian distribution, i.e.

$$b(o \mid S_j) = N(o \mid \mu_j, \sum_j) \tag{8}$$

Where $N(o|\mu,\Sigma)$ denotes a Gaussian density of mean μ and covariance Σ, evaluated at o;

- π =$\{\pi_i\}$, the initial state probability distribution, representing probabilities of initial states, i.e.

$$\pi_i = P[q_1 = S_i] \; 1 \le i \le N \tag{9}$$

with $\pi_i\geqq 0$ and $\sum_{i=1}^{N} \pi_i = 1$.

For convenience, we denote an HMM as a triplet $\lambda = (A, B, \pi)$.

The training of the model, given a set of sequences $\{O_i\}$, is usually performed using the standard Baum-Welch re-estimation [8], which determines the parameters (A, B, π) that maximize the probability P ($\{O_i\}|\lambda$). In this paper, the training procedure is stopped after the convergence of the likelihood. The evaluation step, *i.e.* the computation of the probability $P(O|\lambda)$, given a model λ and a new observation sequence O, is performed using the forward-backward procedure [8]. By using the Hidden Markov Model, we can classify the set of feature points into eyes, nose or other facial feature.

4.2.1 Training

To perform as a recognizer, the model should be trained first. The maximum likelihood training is a well understood technique. However, the iterative maximum likelihood estimation of the parameters only converges to a local optimum, making the choice of the initial parameters of the model a critical issue. The parameters of the HMM used in our system are defined below:

Symbol Sequence: $E = \begin{pmatrix} M^S \\ M^E \end{pmatrix}$

States: q=(q_1=1,$\cdots$,q_t=i,)

HMM Parameter Set: $\lambda = (A, B, \pi)$

Initial State Distribution: $\pi_0^c = P(q_t^c = i)$

State-Transition Probability: $A_{M\times N} = \{a_{i|j,k}{}^c\}$

Observable Symbol Probability: $B_{M\times N} = \{b_i{}^c(o_{t+1})\}$ at state i and time t+1;

Output Probability : P(E|λ)

4.2.2 Recognition

The facial feature recognition is carried out through the computation of viterbi algorithm [9] for the input feature point vector. The parameters of the HMM corresponding to the basic expression are obtained after training.

Through the recognition approach, the facial feature points set have been classified into the follow states:

a. Left Eyes
b. Right Eyes
c. Left Eyebrow
d. Right Eyebrow
e. Nose
f. Mouth

In this way, we can apply all of these points to control the motion of 3D face avatar which will be discussed in the next section.

5. AVATAR MOTION CONTROLLING

For feature points and interpolated feature vertices on 3D generic model, the animation parameter

is obtained by feature point mapping and normalization. The animation parameter of non-feature vertices can be derived by the interpolation of animation parameters (AP) of feature vertices.

5.1 Avatar Pose Controlling

In addition to the motion vectors of tracked face features, mouth position vectors are captured as well. The geometric relationship between mouth and face can be used to estimate the head pose. As shown in Figure 7, when the head rotates from one side to the other side (For example: from the frontal face to the left 45 degree side in the x direction), in the captured 2D image, the mouth moves linearly from the horizontal center to the corresponding position. The procedure has been discussed in our previous papers [10], [11] in detail. According to the procedure, we can use the following equations to estimate the head pose.

$$\alpha = \frac{\pi}{2} \times \frac{2P_M(x) - D_{LR}}{D_{LR}} \tag{10}$$

$$\beta = \frac{\pi}{2} \times \frac{2P_M(y) - D_{TB} - 2D_{MC}}{D_{TB}} \tag{11}$$

In equation (10), D_{LR} is the width of the face region, while in equation (11), D_{TB} is the height of the face region and $P_M(x, y)$ is the mouth position, and these values can be easily obtained by using the face feature detection algorithm. D_{MC} is a constant value that can be obtained from the frontal view face image by experiment. We made

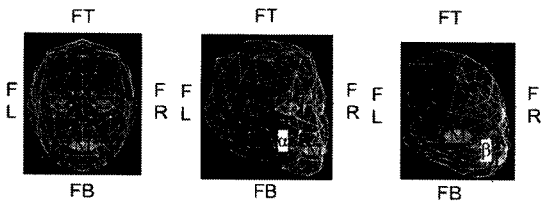


Fig. 7. A Schematic Drawing of Head Pose Estimation Approach: (a) Frontal View; (b) Face Orientation Changed at Horizontal Direction and (c) Face Orientation Changed at Vertical Direction.

(a) (b)

Fig. 8. Avatar Pose Tracking Result: (a) Head Orientation Direction and (b) Avatar Orientation Direction

numerous tests based on the FERET face database to determine the value of D_{TB} which was found to be 7/40. And avatar pose estimation result is as shown in Figure 8.

5.2 Avatar Animation Controlling

The avatar 3D animation vector V of feature vertex i (denoted as V^i) can be derived by the following equation:

$$V^i = \sum_{j=1}^{N} (\omega(d_{i,j}) \cdot V_k^j) \tag{12}$$

Where $d_{i,j}$ is the distance between vertex i and vertex j, $d_{i,j}$, j = 1, 2, ···, N, (e.g. N=3) have been arranged in increasing order. $\omega(d_{i,j})$ is a weight function which will have a large output of weight value for a small input of $d_{i,j}$ as Figure 9 shows.

$$\omega(d_{i,j}) = \frac{d_{i,j'}}{\sum_{j=1}^{N} d_{i,j}}, \text{ where } j' = (N+1) - j \tag{13}$$

The animation previously derived from the

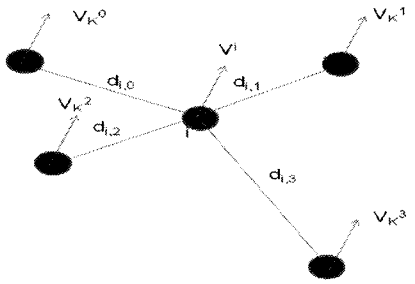


Fig. 9. The Animation Parameter for a Vertex i in Avatar 3D Model.

detected facial feature points, and non-feature points are also added into the feature points list dynamically. The non-feature points are inferred from the feature points based on the updated feature points list. The newly added feature points will influence the non-feature points of the local region, therefore the region of influence on a non-feature point is dynamically changed. While more feature points are considered, the calculation turns out to be more complex. In view of the fact that control points which are further from current vertex pay less influence on the final movement of this vertex, we could use only those features that contribute significantly to the movement of current vertex. In our experiment, less than 5 nearest features points are considered for achieving the satisfactory results.

6. EXPERIMENTAL RESULTS

The system is performed on the setup of regular Pentium-4 PC and an USB web cam video camera, with runs fully automatically from face tracking to avatar controlling. The 3D avatar model consists of 3000 vertices. The modeling process allows us to create an individualized 3D avatar for each subject efficiently. Figure 10 shows several sample frames captured from real time video clip and avatar controlling result. Experiment shows the good visual quality on created expressions with head pose change and mouth open and close.

Our face tracker always starts with mouth

(a) (b)

Fig. 10. Sample Frames of the Real Time Facial Expression and Pose and Avatar Controlling Result: (a) Face Motion and Expression and (b) Avatar Motion Controlling Result.

tracking. So the mouth position from our mouth tracker can provide strong and reliable information about the rough location of face as well as the face motion between two consecutive frames. The robustness of the face tracker relies on the result of the mouth tracker. Since our proposed LCS based method mouth tracker can robustly track the mouth under variable illuminations, the face location and motion information can be effectively inferred from the detected mouth for the face tracker. In general, our system can handle people with glasses [12] well when pupils are not occluded by the glares.

Real time tests have been conducted intensively for a large number of people who performed mo-

tions and expressions in front of the camera. Note that our algorithm requires all the facial features to be visible with respect to the camera for the effective detection. The facial features tracker can tolerate about +- 30 degrees for both tilt and pan angles. When face rotations are beyond these angles, some non-visible features may not be reliably tracked. In this situation, the distortion of avatar controlling may occur due to the large rotation of the head and its non-trivial movement in the depth direction. However, as soon as the facial features became visible after the head moves back, our algorithm is able to grasp the features immediately, and consequently the animation parameters can be reliably derived. The system can works in a fully automatic fashion without any human intervention very well, and it can work for any person.

6.1 LCS Algorithm Analysis

The algorithm analysis procedure can be divided into two steps: the first is algorithm performance speed and the second is its recognition accuracy, for evaluating the algorithm speed, we execute the program on dual processors (Pentium-4 PC 2.0G), precision of face feature points can be regulated by control the face partition number M and N, and the evaluation approach is performed toward 10 persons, and for each we captures 50 images, A *Correct Determination* (*CD*) is that each feature points can be detected with a tolerable position, otherwise it is called a *False Determination* (FD). Table 1 shows the result. The average performance speed can be calculated from the table that is about 0.0179ms, and correction determination rate can reach to 93.6%.

Table 1. LCS Algorithm Eval·uation Result

	Image No.	*PS (ms)*	*CD*
Person 1	50	0.02	48
Person 2	50	0.01	46
Person 3	50	0.03	45
Person 4	50	0.04	47
Person 5	50	0.01	49
Person 6	50	0.009	48
Person 7	50	0.01	47
Person 8	50	0.02	46
Person 9	50	0.03	45
Person 10	50	0.01	47
Total/Avg	500	0.0179	468

PS: Performance Speed

6.2 HMM based Recognition Algorithm Analysis

The input data of the testing procedure is acquired from the upper result. In the procedure, our job is to evaluate the efficiency of HMM classification machine, the input data set is composed of facial feature points that contains mouth, eye, nose etc. HMM classification machine should classify them correctly.

The performance of the HMM, in these modes, is given in Table 2, where the entries in the table are facial feature state error rates for cases in which the feature was unknown state (US), and for cases in which the feature was known state.

And Table 3 shows performance speed and recognition accuracy rate.

From the table, we can conclude that the average performance speed can achieve at 211.72ms for 468 testint set, and accuracy rate can reach to 92%.

Table 2. Error Threshold of HMM Recognition Algorithm

	Training Set		Testing Set	
	UL	*KL*	*UL*	KL
Mouth	0.39	0.16	0.78	0.35
Left Eye	0.24	0.12	0.63	0.54
Right Eye	0.25	0.11	0.64	0.55
Left Eyebrow	0.35	0.14	0.70	0.43
Right Eyebrow	0.35	0.14	0.71	0.42
Nose	1.15	0.85	3.01	1.65

Table 3. HMM Algorithm Evaluation Result

	Input Data Set NO.	*PS (ms)*	*CD*
Mouth	468	189.5	441
Left Eye	468	215.6	435
Right Eye	468	218.9	436
Left Eyebrow	468	221.5	425
Right Eyebrow	468	210.5	428
Nose	468	214.3	416
Avg	468	211.72	430.16

PS: Performance Speed

7. CONCLUSIONS

In this paper, we proposed a system that automatically detect and track human facial feature points, and apply the tracking information to control the 3D avatar motion. In order to detect the facial feature, we have brought forward the LCS detection algorithm, which can detect the feature by classifying the feature pints into 3 states: edge, object and background. And in order to judge face features (mouth, eyes, eyebrows or nose), we employed the HMM based recognition algorithm to classify the feature points. Then finally, the face feature points tracking information is applied to the 3D avatar model to control the avatar pose and animation. For controlling avatar pose, we utilized the mouth position and facial geometric relationship. And for controlling avatar animation, we calculated the feature point movement distance and created the non-feature vertex as new feature points to control the 3D avatar vertex.

The system is demonstrated to work well under various illumination conditions, even with reflections of eye glasses on subjects. The experiments show that it is feasible to control the 3D avatar motion with very limited number of facial features points for extremely low bit rate transmission.

In order to increase the realism and the robustness, large motion detection in the depth direction will be developed in the future.

ACKNOWLEDGMENTS

This research was supported by the MKE (Ministry of Knowledge and Economy), Korea, under the U-Port ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment) (IITA-2009-C1090-0801-0004)

REFERENCES

[1] Haibo Wangy, Chunhong Pan, Christophe Chaillou, Jeremy Ringard, "An Online Face Avatar under Natural Head Movement," IEEE International Conf. on Automatic Face and Gesture Recognition (ECAG08), Amsterdam, Sep. 16, 2008.

[2] Xiaozhou Wei, Zhiwei Zhu, Lijun Yin and Qiang Ji, "A Real Time Face Tracking and Animation System," IEEE International Conf. on Computer Vision and Pattern Recognition Workshop, pp. 71-71, June 27. 02. 2004.

[3] C. Tomasi and T. Kanade, "Detection and Tracking of Point Features," *Carnegie Mellon University Technical Report*, (CMU-CS-91-132), Apr. 1991.

[4] Matthew Turk, Changbo Hu, Rogeria Feris, Farshid Lashkari, Andy Beall, "TLA Based Face Tracking," The 15th International Conference on Vision Interface, May 2002.

[5] Feng Jiao, Stan Li, Heung-Yeung Shum, Dale Schuurmans, "Face Alignment Using Statistical Models and Wavelet Features," *Proc. of IEEE CVPR*, 2003.

[6] Z. Zhang, "Feature-based facial expression recognition: Experiments with a multi-layer perception," Technical Report INRIA, pp. 335, 1998.

[7] P. Viola and M. Jones, "Robust Real-time Object Detection", *Proc. of 2nd International*

Workshop on Statistical and Computational Theories of Vision-Modeling, Learning, Computing and Sampling, pp. 1-25, July 2001.

[8] L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. of IEEE*, Vol.77, No.2, pp. 257-286, 1989.

[9] Lou and H.-L. "Implementing the Viterbi algorithm," *IEEE Journal on Signal Processing Magazine*, Vol.12, No.5, pp. 42-52, Sept. 1995.

[10] Li Wei, E. J. Lee and S. Y. Ok, "Automatic Camera Pose Determination from a Single Face Image," *Journal of Korea Multimedia Society*, Vol.10, No.12, pp. 1566-1576, Dec. 2007.

[11] Li Wei, E.J. Lee, "Human Head Mouse System Based on Facial Gesture Recognition," *Journal of Korea Multimedia Society*, Vol.10, No.12, pp. 1591-1600, Dec. 2007.

[12] Z. Zhu, K. Fujimura, Q. Ji, "Real-Time Eye Detection and Tracking Under Various Light Conditions and Face Orientations", 2002 *ACM SIGCHI Symposium on Eye Tracking Research & Applications*, New Orleans, LA, 2002.

Li Wei

Li Wei received his B. S. in Dalian University of Light Industry in China (2002 - 2006), and M. S in TongMyong University, Korea (2006-2008), and now (2008 - 2009) his is a Ph. D. student of in Korea. His main research interests are in image processing computer vision Biometrics and face recognition.

Eung-Joo Lee

Eung-Joo Lee received his B. S.,M., S. and Ph. D. in Electronic Engineering from Kyungpook National University, Korea, in 1990, 1992, 1996, respectively. In March 1997, he joined the Department of Information Communication Engineering of Tongmyong University, Busan, Korea, as a professor. His main research interests are in image processing, computer vision, and Biometrics.