

# Recognition of Human Facial Expression in a Video Image using the Active Appearance Model

Gyeong-Sic Jo\* and Yong-Guk Kim\*

**Abstract**—Tracking human facial expression within a video image has many useful applications, such as surveillance and teleconferencing, etc. Initially, the Active Appearance Model (AAM) was proposed for facial recognition; however, it turns out that the AAM has many advantages as regards continuous facial expression recognition. We have implemented a continuous facial expression recognition system using the AAM. In this study, we adopt an independent AAM using the Inverse Compositional Image Alignment method. The system was evaluated using the standard Cohn-Kanade facial expression database, the results of which show that it could have numerous potential applications.

**Keywords**—Active Appearance Model, Facial Expression Recognition, Image Alignment Method

## 1. INTRODUCTION

Facial expression recognition is a crucial method of inferring human emotions. Facial emotions are basically categorized into six facial expressions (surprise, fear, sadness, anger, disgust, happiness). The process flow of the present system is shown in Figure 1. Facial-expression images are captured from a web camera. The active appearance model includes various parameters of shape and appearance. With these images, an AAM *instance* is generated, and its emotion is classified by EFM.

In Section 2, the AAM is reviewed; in Section 3, the process of the EFM classifier is described; in Section 4, the performance of the system is evaluated using the Cohn-Kanade facial expression database; and, finally, the work is summarized and the conclusion presented in Section 5.

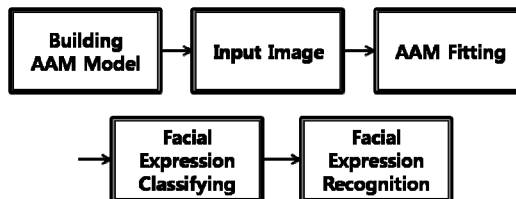


Fig. 1. Block diagram of the facial expression recognition system

※ This work was supported by the Korean Research Foundation (313-2007-2-D00724).

Manuscript received March 3, 2010; accepted March 24, 2010.

Corresponding Author: Yong-Guk Kim

\* Dept. of Computer Engineering, Sejong University, Seoul, Korea (ykim@sejong.ac.kr)

## 2. ACTIVE APPEARANCE MODEL

The AAM was first proposed in [2]. Typical applications of the AAM involved the modeling and recognizing of human faces. However, the AAM was also effective in modeling other objects. In addition, the model can be transferred to other applications. The AAM, though initially proposed for modeling the human face, can be useful for modeling facial expressions [1]. The main purpose of the AAM is to build a new model instance by finding the best matched parameters between the input images and a given model with a fitting algorithm. The fitting algorithm, which is of a non-linear mode, iterates itself until the parameters of both shape and appearance satisfy particular values. For instance, when the parameter of shape is measured, one can fit an input image onto the coordinate frame of the model. After such a match, the gap between the instance of the model and the pixels within the shape of the input image can be measured. This gap would be applied to the fitting algorithm in order to update the parameters. Iterating this process would make optimized parameters as the fitting algorithm converged to a certain value. We have adopted an Inverse Compositional Image Alignment, one of its fitting algorithms used in this system, as shown in section 2.2.

### 2.1 Model Instance

Firstly, the shape of the AAM is created by combining the vectors, which are made by marking the points on the images by hand.

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^n p_i \mathbf{s}_i \quad (1)$$

In equation (1),  $p_i$  means the parameters of shape,  $s_0$  indicates the base shape, and  $s_i$  represents the shape vectors. The eigenvectors for the shape can be obtained using Principal Component Analysis (PCA). They are the  $n$  eigenvectors which correspond to the  $n$  largest eigenvalues. Before applying PCA, the AAM usually uses *Procrustes* analysis in order to normalize landmark points that are marked manually [1].

$$A(\mathbf{x}) = A_0(\mathbf{x}) + \sum_{i=1}^m \lambda_i A_i(\mathbf{x}) \quad (2)$$

where  $\lambda_i$  indicates the appearance parameters,  $A_i$  represents the appearance vectors, and  $A_0$  is the base appearance. After finding both the shape and appearance parameters, the AAM instance is generated by locating each pixel of appearance on the inner side of the current shape with a piecewise affine warp. A model instance is indicated, as in equation (3).

$$M(\mathbf{W}(\mathbf{x}; \mathbf{p})) = A(\mathbf{x}) \quad (3)$$

The parameters of both shape and appearance are obtained by a fitting algorithm. Figure 2 shows the process of creating the model instance.

## 2.2 Inverse Compositional Image Alignment

The aim of image alignment is to find the location of a fixed template image on an input image. Lucas and Kanade first proposed the Image Alignment Method [3]. Their algorithm is designed to locally align a fixed template on the image, as shown by equation (4).

$$\sum_x [A_0(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p}))]^2 \tag{4}$$

The parameter  $\mathbf{P}$  plays a role in minimizing errors between the fixed template  $A_0(\mathbf{x})$  and the input image  $I(\mathbf{x})$ . In  $\mathbf{x} = (x, y)^T$  is the coordinator of the pixel. The parameter  $\mathbf{P}$  was linear, although  $I(\mathbf{x})$  was non-linear. Thus, this formula is a non-linear optimization problem. To solve this problem linearly, the Lucas-Kanade algorithm assumes that  $\mathbf{P}$  is already known. Then, the algorithm increases the difference of  $\mathbf{P}$  repeatedly, marked  $\Delta \mathbf{p}$  in (5).

$$\sum_x [A_0(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p} + \Delta \mathbf{p}))]^2 \tag{5}$$

$\mathbf{P}$  is increased by adding  $\Delta \mathbf{p}$  to  $\mathbf{P}$ . When changing the value of  $\mathbf{P}$ , the performance of the Lucas-Kanade algorithm is very slow because three parameters - Jacobian, gradient image, and Hessian Matrix - have to be computed over and over again. To improve this performance, the Forwards Compositional Image Alignment method is introduced. In this algorithm,  $\mathbf{P}$  is changed by combining the  $\mathbf{W}(\mathbf{x}; \mathbf{p})$  and  $\mathbf{W}(\mathbf{x}; \Delta \mathbf{p})$ , as given in equation (6).

$$\sum_x [A_0(\mathbf{x}) - I(\mathbf{W}(\mathbf{W}(\mathbf{x}; \Delta \mathbf{p}); \mathbf{p}))]^2 \tag{6}$$

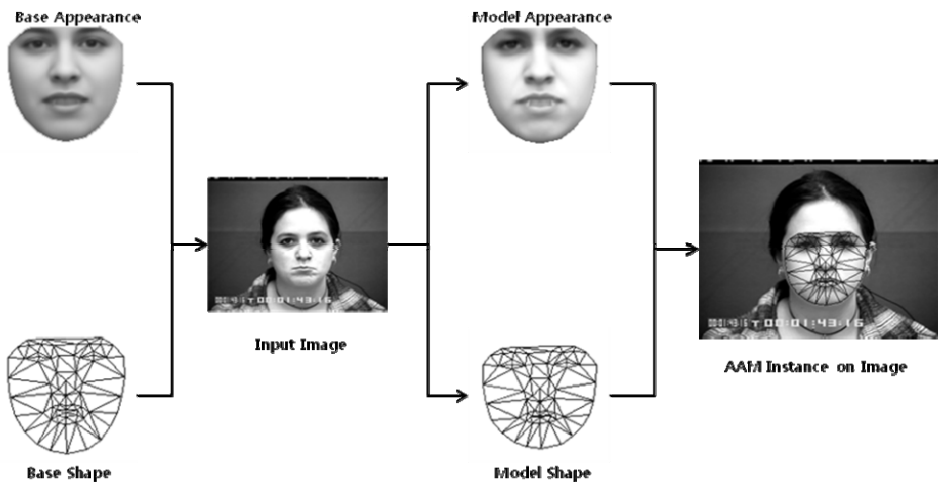


Fig. 2. Generation of an AAM instance



Fig. 3. Real-time face tracking using ICIA

When using the Forwards Compositional Image Alignment method, one does not need to compute Jacobian every time, since the algorithm can compute Jacobian in  $(x;0)$ . In this paper, the Inverse Compositional Image Alignment method, in which the positions of the input image and the template image are exchanged, is adopted. The ICIA can be formulated as equation (7).

$$\sum_x [I(\mathbf{W}(x; \mathbf{p})) - A_0(\mathbf{W}(x; \Delta \mathbf{p}))]^2 \quad (7)$$

The main advantage of this method is that the speed of updating parameters  $\mathbf{P}$  can be very rapid, since Jacobian and Gradient Image are calculated at  $A_0(x)$ . Once those values have been acquired in the initial stage, they can be used whenever one wishes to update a new warp parameter. Figure 3 shows three images of a face tracked using this method.

### 3. EFM CLASSIFIERS

This section introduces an EFM classifier, by which the discriminative features between facial expressions are determined. The EFM is introduced to improve the generalization ability of standard Fisher Linear Discriminant (FLD)-based methods.

#### 3.1 EFM Algorithm

Let  $Y$  be a random vector representing the lower dimensional feature. Let  $w_1, w_2, \dots, w_L$  and  $N_1, N_2, \dots, N_L$  denote the classes and the number of images within each class, respectively. Let  $M_1, M_2, \dots, M_L$  and  $M$  be the mean of the classes and the grand mean. The within- and between-class co-variance matrices  $\Sigma w$  and  $\Sigma b$  are defined as follows:

$$\Sigma w = \sum_{i=1}^L P(w_i) E\{(Y - M_i)(Y - M_i)' | w_i\} \quad (8)$$

$$\Sigma b = \sum_{i=1}^L P(w_i) (M_i - M)(M_i - M)' \quad (9)$$

The EFM first diagonalizes the within-class co-variance matrix  $\Sigma w$ .

$$\sum w \Xi = \Xi \Gamma \text{ and } \Xi' \Xi = I, \tag{10}$$

$$\Gamma^{-1/2} \Xi' \sum w \Xi \Gamma^{-1/2} = I, \tag{11}$$

where  $\Xi, \Gamma$  are the eigenvector and the diagonal eigenvalue matrices of  $\sum w$ , respectively. The EFM then proceeds to compute the between-class co-variance matrix as follows:

$$\Gamma^{-1/2} \Xi' \sum b \Xi \Gamma^{-1/2} = K_b, \tag{12}$$

The EFM diagonalizes the new between-class co-variance matrix  $K_b$ .

$$K_b \Theta = \Theta \Delta \text{ and } \Theta' \Theta = I, \tag{13}$$

where  $\Theta, \Delta$  are the eigenvector and the diagonal eigenvalue matrices of  $K_b$ , respectively. The overall transformation matrix of the EFM is finally defined as follows:

$$T = \Xi \Gamma^{-1/2} \Theta \tag{14}$$

## 4. EXPERIMENTS AND PERFORMANCE

Four types of facial expression were used for the purposes of the present study: neutral, sad, happy, and surprised. The AAM model was established using 498 facial images, and each image was marked by 68 landmark points. The EFM model was set up with 54 ‘neutral’, 50 ‘sad’, 50 ‘happy’, and 50 ‘surprised’ images. The experiments were divided into two types of evaluation: one evaluated how correctly each model classified the images of the facial expressions, and the other tested how exactly the system analyzed the sequential images, including various facial expressions.

### 4.1 EFM Algorithm

For the performance evaluation, the 5-Cross Validation method was employed, whereby 1/5 facial expression images were taken as a test set, while the rest were used as the EFM train set. Thus, five tests were conducted, the results of which are shown in Tables 1 and 2. The results

Table 1. Recognition Results

|           | Test Image No. | Success No. | Rate (%) |
|-----------|----------------|-------------|----------|
| Neutral   | 54             | 46          | 85.1     |
| Sad       | 50             | 41          | 82.0     |
| Happy     | 50             | 46          | 92.0     |
| Surprised | 50             | 45          | 90.0     |
| Total     | 204            | 178         | 87.2     |

Table 2. Recognition Results

|           | Neutral | Sadness | Happiness | Surprise |
|-----------|---------|---------|-----------|----------|
| Neutral   | 46      | 6       | 0         | 2        |
| Sad       | 8       | 41      | 0         | 1        |
| Happy     | 2       | 2       | 46        | 0        |
| Surprised | 5       | 0       | 0         | 45       |

show that the worst case concerned the neutral or sadness facial expressions. In the confusion matrix, note that the range of error between the neutral and sadness case was considerable. It seems that it is rather difficult to distinguish between two subtle facial expressions.

#### 4.2 Continuous Expression Image Analysis

Since the Cohn-Kanade database consists of sequential images of a facial expression, it was possible to use it to test the proposed system for continuous facial expression recognition cases. For instance, Figure 4 shows how the ‘surprised’ expression evolves from the neutral one, in particular, between image sequences 4 and 7. Here, the horizontal axis represents the sequence’s number of images, whereas the vertical axis indicates the EFM distance between the facial expression ranks. After sequence 7, the ‘happy’ expression becomes a dominant one, showing a large distance from the other expressions. The system can process 15 frames per second.

Figure 5 shows that the ‘sad’ expression shows a distance from the two expressions (‘happy’ and ‘surprised’) as time goes by, ranging from sequence 2 to 6, while the ‘neutral’ expression drifts from the bottom in an upward direction. However, note that the distance between the ‘neutral’ expression and the ‘sad’ one is not very great even for sequence 10, indicating that the two expressions are, in a way, similar. This observation confirms the result shown in Table 2, in which the system appears to be confusing two expressions.

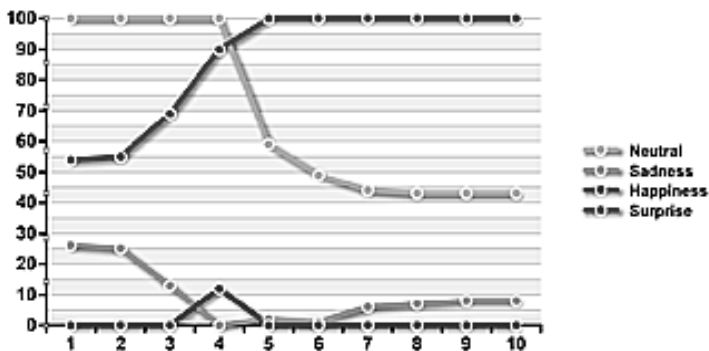


Fig. 4. Block diagram of the facial expression recognition system

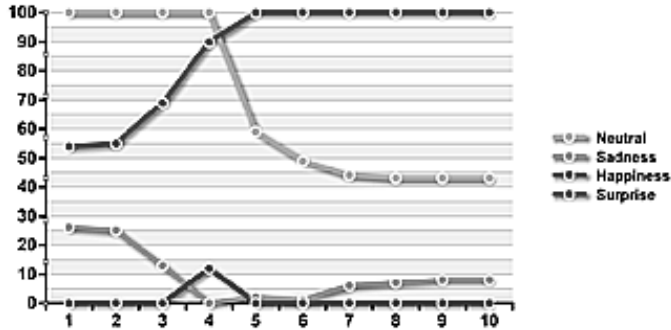


Fig. 4. Block diagram of the facial expression recognition system

## 5. CONCLUSION AND DISCUSSION

This paper describes how the proposed real-time facial expression recognition system, using AAM and EFM, was implemented, and how two tests were conducted in order to examine the performance of the system. The results suggest that the system carried out facial expression recognition very well, and that it also operates in the continuous facial expression recognition task. As it is known that facial expressions tend to be accompanied by motion of the head, the authors are working on a project in which the head tracker is combined with the present facial expression tracker.

## REFERENCES

- [1] Matthews, I., Baker, S., *Active Appearance Models Revisited*, International Journal of Computer Vision, pp.135-164, 2004.
- [2] Edwards, G.J., Taylor, C.J., and Cootes, T.F., *Interpreting Face Images using Active Appearance Models*, In Proc. of the International Conference on Automatic Face and Gesture Recognition, pp.300-305, June, 1998.
- [3] Lucas, B., and Kanade, T., *An iterative image registration technique and its application to stereo vision*, In Proc. of the International Joint Conference on Artificial Intelligence, pp.674-679, 1981.



### Gyeong-Sic Jo

Jo received his BS and MS in Computer Engineering from Sejong University in 2004 and 2007, respectively. At present, he is a Ph.D. candidate, and is focusing on human facial expression recognition and machine learning.



**Yong-Guk Kim**

Kim received his BS and MS in Electrical Engineering from Korea University in 1982 and 1984, respectively. His work experience includes research posts at LG Electronics from 1984 to 1985 and at KT from 1986 to 1990. His Ph.D., which he received from Cambridge University in 1997, concerned a visual surface representation model. From 1998 to 2001, he was a research associate of the Smith-Kettlewell Vision Research Institute in the USA. He has been with the Department of Computer Engineering, Sejong University since 2001. His research interests are facial expression recognition, computational vision, and Human-Computer Interaction.