

# Active Appearance Model과 EFM을 이용한 실시간 얼굴 표정 인식 시스템

조경식<sup>○</sup> 김용국 이양복  
세종대학교

nopnse@naver.com<sup>○</sup>, ykim@sejong.ac.kr, yangbok@empal.com

Real-time Facial Expression recognition System using  
Active Appearance Model and EFM

Kyoung-Sic Cho<sup>○</sup>, Y.G. Kim, Y.B. Lee

School of Computer Engineering, Sejong University, Seoul, Korea

## 요 약

본 논문에서는 Active Appearance Model과 EFM을 기반으로 하는 실시간 표정인식 시스템을 설명한다. AAM은 얼굴추적, 얼굴인식 그리고 물체인식과 같은 시스템에 널리 사용되어 지고 있다. 시스템에 사용된 AAM은 Inverse Compositional Image Alignment를 적용한 Independent AAM으로서 fitting 속도가 빨라 실시간 시스템에 매우 효과적이다. 시스템의 성능 평가는 Cohn-Kanade Image DB의 표정영상과 연속영상을 사용하여 실시 하였다.

## 1. 서 론

얼굴표정 인식은 사람의 감정을 유추할 수 있는 중요한 수단으로서 Vision based HCI에 중요한 위치를 차지하고 있다. 사람의 얼굴 표정은 기본적으로 6개의 표정(놀람, 두려움 슬픔, 화남, 역겨움, 즐거움)으로 나누어진다. AAM을 이용한 인식 시스템은 보통 shape parameter와 appearance parameter를 다양한 classifier넣어서 분류 한다[1]. 시스템의 전체 구조는 Fig.1과 같다. 트레이닝 영상으로 AAM 모델과 EFM 모델을 만들고 Web-cam으로 촬영한 영상이 입력되면 AAM Model instance를 만들고 EFM으로 분류를 하여 최종적으로 표정인식을 수행하고 인식이 끝났으면 다시 Web-cam으로 영상을 받아 들인다.

본 논문의 구성은 다음과 같다. 2장에서는 independent AAM을 설명하고 3장에서는 EFM Classifier에 관하여 설명한다. 4장에서는 표정인식 시스템의 성능평가와 연속 영상의 표정인식 결과를 보여주고, 마지막 5장에서는 결론을 맺는다.

## 2. Independent Active Appearance Models(AAMs)

AAMs는 [2]에서 처음 제안되었다. 대부분의 AAMs Application이 얼굴을 모델링 하는데 사용되지만 다른 객체를 모델링할 때에도 매우 효과적이다[3][4]. 또한 하나의 모델은 다른 Application에서도 사용 가능한데 예를 들어 얼굴인식을 하기 위해 만들어진 AAMs는 표정인식이나 얼굴추적 등에서도 사용할 수 있다[6]. AAMs은 입력영상과 모델이 최대한 잘 매치되도록 하는 파라미터를 Fitting Algorithm으로 찾아서 새로운 모델 인스턴스를 생성하는 것이 목적이다. AAMs에서 사용하

는 Fitting Algorithm은 비선형으로서 Shape과 Appearance의 파라미터가 특정 임계값을 만족할 때 까지 반복적으로 갱신한다. 예를 들어 현재 Shape의 파라미터가 측정 되었으면 모델 좌표 프레임위에 입력영상을 맞추고 모델 인스턴스와 입력영상에서 shape 내에 있는 픽셀들의 오차영상을 구해 이 오차영상을 Fitting Algorithm에 적용 하여 파라미터를 갱신한다. 이 과정을 반복면 최적화된 파라미터를 구할 수 있다. 본 시스템에서 사용된 Fitting Algorithm은 Inverse Compositional Image Alignment로서 이번 장의 2절에 자세히 설명하겠다.

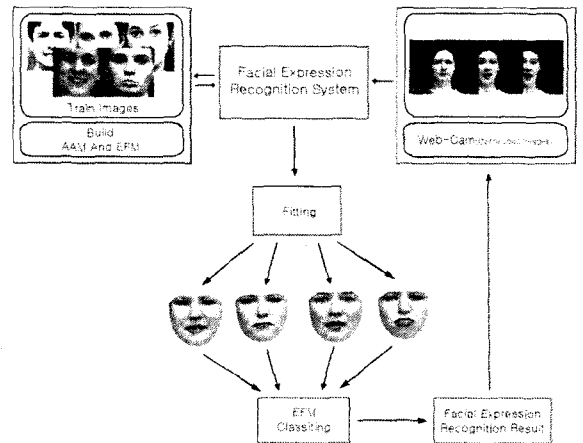


그림. 1 표정 인식 시스템

### 2.1 Model Instance

Independent AAM의 Shape은 영상에 직접 표기한 Landmark의 좌표를 벡터화 시키고 이 벡터를 선형조합하여 생성한다.

$$s = s_0 + \sum_{i=1}^n p_i s_i \quad (1)$$

여기서  $p_i$ 는 shape파라미터 이고,  $s_i$ 는 shape vector 그리고  $s_0$ 는 base shape 이다. Shape vector는 Landmark 좌표들을 PCA하여 가장 큰 eigenvalue와 대응하는 eigenvectors 이다. AAM에서는 PCA 하기 전에 일반적으로 Procrustes analysis를 사용하여 Landmark들을 정규화를 한다[1].

Independent AAM의 appearance는 base mesh  $s_0$  내에 있는 pixel로 정의 된다. Appearance도 shape과 마찬가지로 선형조합으로 생성한다.

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

여기서  $\lambda_i$ 는 appearance파라미터 이고,  $A_i$ 는 appearance vector이다. 마지막으로  $A_0$ 는 base appearance 이다. 선형 조합을 위해 appearance도 PCA를 하여 가장 큰 eigenvalue에 대응하는 eigenvector를 사용한다.

AAM의 instance는 적절한 Shape parameter  $p_i$ 와 appearance parameter  $\lambda_i$ 가 구해지면 Appearance의 각 픽셀들을 piece affine warp을 이용하여 현재 shape의 내에 위치 시킴으로서 생성된다. Model instance는 다음으로 표기한다.

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

Shape parameter  $p_i$ 와 appearance parameter  $\lambda_i$ 는 Fitting Algorithm을 사용하여 구하는데 이것에 대한 것은 다음절에서 설명하도록 하겠다. 다음의 Fig.2는 Model Instance 생성과정의 보여준다.

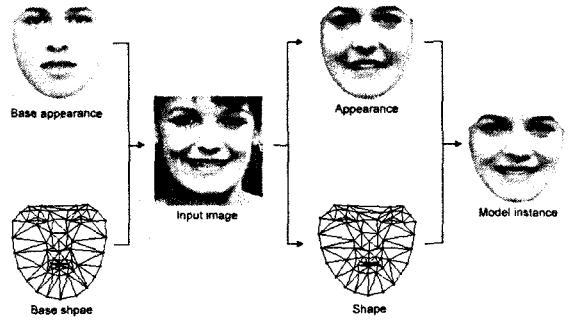


그림. 2 AAM instance

### 2.2 ICIA(Inverse Compositional Image Alignment)

Image Alignment의 목적은 입력 영상에서 고정된 템플릿 영상의 위치를 찾는 것이다. 최초의 영상정렬 알고리즘은 Lucas와 Kanade가 소개하였다[7]. 이들이 소개한 영상정렬은 국지적으로 입력영상이 고정된 템플릿에 정렬되게 하는 것으로서 다음과 같이 표기 된다.

$$\sum_x [A_0(x) - I(W(x;p))]^2 \quad (4)$$

식(4)에서 warp 파라미터  $\mathbf{p}$ 는 고정된 템플릿  $A_0(x)$ 와 입력영상  $I(x)$  사이에 오차를 최소화 하는 역할을 한다. 픽셀 좌표  $\mathbf{x} = (x, y)^T$ 에서 픽셀 값  $I(x)$ 는 비선형이기 때문에  $\mathbf{p}$ 가 선형일지라도 이것은 비선형 최적화 문제이다. 따라서  $\mathbf{p}$ 를 선형화 문제로 해결하기 위하여 Lucas-Kanade 알고리즘에서는  $\mathbf{p}$ 를 이미 알고 있다고 가정하고 식(5)에서 보이는  $\Delta \mathbf{p}$ 를 반복적으로 증가시킨다.

$$\sum_x [A_0(x) - I(W(x;p + \Delta p))]^2 \quad (5)$$

$\mathbf{p}$ 의 갱신은 위와 같이  $\Delta \mathbf{p}$ 를  $\mathbf{p}$ 에 더하는 것으로 이루어 진다. Lucas-Kanade 알고리즘은 입력영상  $I(x)$ 에서 warp parameter  $\mathbf{p}$ 를 갱신할 때 필요한 Jacobian, gradient image 그리고 Hessian Matrix를 반복해서 계산하기 때문에 속도가 매우 느리다. 이를 보완하기 위해 고안된 영상 정렬 알고리즘이 Forwards Compositional Image Alignment이다. 이 알고리즘에 warp parameter  $\mathbf{p}$ 는 식(6)에서 보는 것과 같이  $W(x;p)$ 를  $W(x;\Delta p)$ 로 조합하여 갱신한다.

$$\sum_x [A_0(x) - I(W(W(x;\Delta p);p))]^2 \quad (6)$$

Forwards Compositional Image Alignment가 보완된 점은 Jacobian을  $(\mathbf{x}; \mathbf{0})$ 에서 계산하기 때문에 갱신의 때 반복마다 jacobian을 계산할 필요가 없다는 것이다.

본 논문에서는 Forwards compositional Image Alignment에 입력영상과 템플릿 영상의 역할을 역으로 하는 Inverse Compositional Image Alignment를 사용한다. 식(7)은 Inverse Compositional Image Alignment를 나타낸 것이다.

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

이 영상정렬의 장점은 Jacobian과 Gradient Image를  $A_0(\mathbf{x})$ 에서 계산하기 때문에 한번만 계산해 놓으면 warp parameter  $\mathbf{P}$ 를 갱신할 때 계속 사용할 수 있어 갱신 속도가 매우 빠르다는 점이다. Fig. 3은 ICIA를 적용한 AAM으로 실시간 얼굴 추적에 한 것이다.



그림. 3 ICIA를 이용한 실시간 얼굴 추적

### 3 EFM 분류기

이번 장에서는 축소된 영상 공간의 특징을 결정하는 EFM 분류기를 설명한다. EFM은 표준 FLD(Fisher linear discriminant) 기반의 방법을 성능을 향상시키기 위해 소개 되었으며 최초의 EFM은 차원 축소를 위한 PCA(Principle component analysis)에 적용 되어 축소된 PCA 하위공간을 식별하는데 사용 되었다.

#### 3.1 EFM 알고리즘

다음에  $Y$ 는 보다 낮은 차원의 특징을 표현하는 랜덤 벡터를 표현 하고  $w_1, w_2, \dots, w_L$ 와  $N_1, N_2, \dots, N_L$ 는 클래스와 각 클래스 내의 영상 수를 나타내며  $M_1, M_2, \dots, M_L$ 과  $M$ 은 각 클래스의 평균과 전체 평균을 나타낸다. 이들을 가지고 다음과 같이 클래스 내 분산  $\Sigma_w$ 와 클래스 간 분산  $\Sigma_b$ 를 다음과 같이 정의 한다.

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

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

EFM은 최초 다음과 같이 클래스 내 분산 행렬  $\Sigma_w$ 을 다음과 같이 대각화 한다.

$$\Sigma_w \Xi = \Xi \Gamma \text{ and } \Xi' \Xi = I, \quad (10)$$

$$\Gamma^{-1} \Xi' \Sigma_w \Xi \Gamma^{-1} = \Lambda, \quad (11)$$

여기서  $\Xi, \Gamma$ 은 고유 벡터이고  $\Sigma_w$ 는 고유치의 대각선 행렬이다. 식(10), (11)을 가지고 EFM은 식(12) 같이 클래스간 분산 행렬을 계산하고 :

$$\Gamma^{-1} \Xi' \Sigma_b \Xi \Gamma^{-1} = K, \quad (12)$$

식(12)로 구해진 새로운 클래스 간 분산 행렬  $K$ ,는 다시 식(13)으로 대각화 되어 진다.

$$K, \Theta = \Theta \Delta \text{ and } \Theta' \Theta = I \quad (13)$$

식(13)에  $\Theta, \Delta$ 는  $K$ ,의 고유 벡터와 고유치에 대각선 행렬이다. EFM에 전체 변환 행렬은 다음과 같이 정의 된다.

$$T = \Xi \Gamma^{-1} \Theta \quad (14)$$

### 4 실험 및 결과

본 논문에서 사용된 얼굴표정은 Neutral, Sadness, Happiness 그리고 Surprise이다. AAM의 모델은 498장의 얼굴 표정 영상에 68개의 Landmark를 손으로 직접 표기하여 만들었고 EFM 모델은 Neutral 영상 54장, Sadness 영상 50장, Happiness 영상 50장 그리고 Surprise 영상 50장으로 만들었다. 실험은 각 모델이 각각의 표정을 담고 있는 영상을 얼마나 잘 분류하는지 알아보기 위한 성능 평가와 무표정으로 시작하여 각각의 표정으로 변화하는 연속된 영상으로 미세한 표정변화를 얼마나 잘 분석하는지를 알아보는 두 가지의 실험으로 이루어 졌다. 실험에 사용된 표정인식 시스템은 초당 15frame을 인식할 수 있다.

본 실험에는 RAM 1GB를 탑재한 Pentium 4.3GHz의 PC환경에 Visual Studio 6.0를 사용하였다.

#### 4.1 성능 평가

성능 테스트는 시스템에서 EFM모델을 만들 때 사용할 각각의 표정영상을 1/5은 test set으로 사용하고 나머지 4/5는 EFM train set으로 나누는 Cross Validation을 사용하여 실시하였다. 따라서 성능 평가는 총 5회를 실시하였고 각각의 결과를 종합한 것은 다음 table 1, 2와 같다.

표 1. 인식 결과

Test Image No.	Success No.	Rate(%)
Neutral	54	85.1
Sadness	50	82.0
Happiness	50	92.0
Surprise	50	90.0
Total	204	87.2

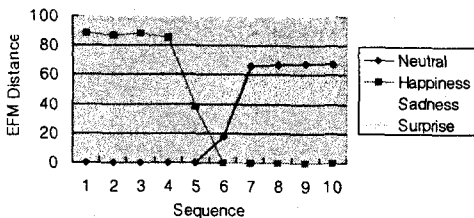
표 2. Confusion Matrix

	Neutral	Sadness	Happiness	Surprise
Neutral	46	6	0	2
Sadness	8	41	0	1
Happiness	2	2	46	0
Surprise	5	0	0	45

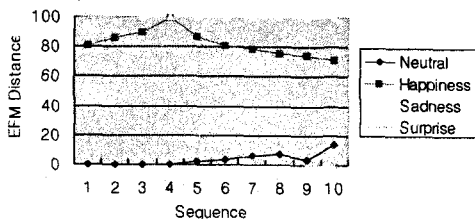
인식 결과를 보면 사람의 유관으로도 비슷하게 보이는 Neutral과 Sadness영상에 대한 인식 결과가 가장 좋지 않았으며, Confusion Matrix를 보았을 때도 Neutral과 Sadness영상이 상호간에 오차를 많이 유발하였다. 이는 얼굴 표정의 특징을 가장 크게 보여주는 입과 눈의 모양이 유사하기 때문이다.

#### 4.2 연속적인 표정 영상 분석

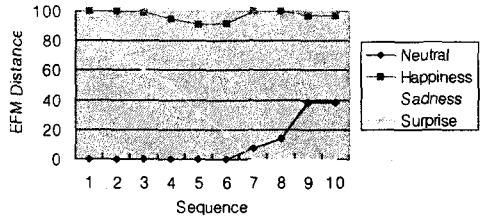
앞서 우리는 데이터 별 표정인식 결과를 알아 보았다. 이번에는 한 사람이 무표정으로 시작하여 각 표정으로 변하는 연속된 영상의 표정인식 결과를 알아본다.



그래프 1. Recognition result of happiness



그래프 2. Recognition result of surprise



그래프 3. Recognition result of sadness

Graph를 보면 시스템이 영상에 담겨있는 각각의 표정 변화대로 EFM distance를 조금씩 줄여감과 동시에 시작 표정인 Neutral의 distance를 벌려가고, 나머지 2표정은 Distance가 벌어진 상태를 유지함을 볼 수 있다. 이는 시스템이 영상에 담겨있는 표정의 흐름을 정확히 분류하고 있다는 것을 보여준다.

#### 5 결론

본 논문에서는 AAM과 EFM을 이용한 실시간 표정인식 시스템을 어떻게 구현하는지를 보여주었으며, 시스템의 성능을 알아보기 위한 2가지 실험을 하였다. 실험의 결과로 우리는 시스템이 표정인식뿐만 아니라 연속영상의 표정의 흐름까지 우수하게 처리함을 볼 수 있었다.

시스템은 아직까지 Single-user에 대한 표정인식을 하고 있는데, 이를 multi-user로 보완 한다면 시스템은 다양한 application에 적용하여 사용 할 수 있을 것이다.

#### 참고문헌

1. I. Matthews, S. Baker "Active Appearance Models revisited" International Journal of Computer vision, pages 135-164, 2004.
2. G. J. Edwards, C. J. Taylor and T. F. Cootes "Interpreting Face Images using Active Appearance Models", In Proc. International Conference on Automatic Face and Gesture Recognition, pages 300-305, June 1998.
3. M. Jones and T. Poggio. "Multidimensional morphable models" In Proceeding of the IEEE International Conference on Computer Vision, pages 683-688, 1998.
4. S. Sclaroff and J. Isidoro. Active Blobs: "region-based, deformable appearance models" Computer vision and Image Understanding, 89(2/3):197-255, Feb. 2003.
5. S. Baker, I. Matthews "Lucas-Kanade 20 Years on: A Unifying Framework Part1", International Journal of Computer Vision, page 221-255, 2004.
6. A. Lanitis, C.J.Taylor, and T.F.Cootes. Automatic interpretation and coding of face images using flexible models. IEEE Transactions on Pattern

Analysiss and Machine Intelligence, 19(7):742-756,  
1997.

7. B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In Proceedings of the International Joint Conference on Artificial Intelligence, pages 674-679, 1981.