

얼굴인식을 위한 3D Active Appearance Model

3D Active Appearance Model for Face Recognition

조경식, Kyoung-Sic Cho, 김용국, Yong-Guk Kim
세종 대학교 컴퓨터 공학과

요약 Active Appearance Models 은 객체의 모델링에 널리 사용되며, 특히 얼굴 모델은 얼굴 추적, 포즈 인식, 표정 인식, 그리고 얼굴 인식에 널리 사용되고 있다. 최초의 AAM 은 Shape 과 Appearance 가 하나의 계수에 의해서 만들어 지는 Combined AAM 이었고, 이후 Shape 과 Appearance 의 계수가 분리된 Independent AAM 과 3D 를 표현할 수 있는 Combined 2D+3D AAM 이 개발 되었다. 비록 Combined 2D+3D AAM 이 3D 를 표현 할 수 있을지라도 이들은 공통적으로 2D 영상을 사용하여 모델을 생산한다. 본 논문에서 우리는 stereo-camera based 3D face capturing device 를 통해 획득한 3D 데이터를 기반으로 하는 3D AAM 을 제안한다. 우리의 3D AAM 은 3D 정보를 이용해 모델을 생산하므로 기존의 AAM 보다 정확한 3D 표현이 가능하고 Alignment Algorithm 으로 Inverse Compositional Image Alignment(ICIA)를 사용하여 빠르게 Model Instance 를 생산할 수 있다. 우리는 3D AAM 을 평가하기 위해 stereo-camera based 3D face capturing device 로 촬영해 수집한 한국인 얼굴 데이터베이스[9]로 얼굴인식을 수행하였다.

핵심어: AAM, 3D AAM, stereo-camera based 3D face capturing, Raw 3D Data

1. Introduction

Active Appearance Models (AAMs)는 [1]에서 처음 제안되었고 [2]과 [3]에서 개선되었다. AAMs는 보통 얼굴을 모델링 하는데 많이 사용되지만 다른 객체를 모델링할 때에도 효과적이다. 얼굴 모델은 주로 얼굴 추적, 포즈 인식, 표정 인식, 그리고 얼굴 인식에 사용될 수 있다. [1]과 [2]의 AAMs는 2D영상에 정보만을 이용해 구축된다. 따라서 모델의 Shape과 Appearance는 2D로 표현된다. [3]에서는 [2]를 확장 시켜 2D를 3D로 표현 하는 Combined 2D+ 3D AAM를 제안 하였다. 2D에서3D로의 확장 방법은 2D AAM 으로 연속 영상 속의 얼굴을 추적하여 2D AAM shape vector를 계산하고 [10]의 non-rigid structure-from-motion algorithms를 사용해서 3D linear shape mode로 변환 시키는 것이다.

비록 Combined 2D+ 3D AAM이 3D를 표현 할 수 있을지라도 앞서 말한 것과 같이 트레이닝 데이터로 2D 정보를 사용한다. 따라서 이것은 본질적으로 3D 데이터를 사용해 3D를 표현하는 것보다 효과적이지 않다고 볼 수 있다. 본 논문에서 우리는 stereo-camera based 3D face capturing device 를 통해 획득한 3D 데이터를 확장된 AAM 에 적용해 효과적으로 3D모델을 표현하는 3D AAM을 제안한다. 제안된 알고리즘은 2D AAM에 영상의 밀도 차와 3D Geometry를 이용하여 계산한 depth Model을 추가하는 방법을 사용한다. 이 알고리즘의 장점은 첫째로 모델 표현을3D로 하기 때문에 2D AAM 보다 모델을 자유롭게 표현할 수 있다는 것이고 둘

째는 3D를 표현함에도 Inverse Compositional Image Alignment(ICIA)를 사용하기 때문에 Fitting 속도는 2D AAM과 거의 같다는 것이다.

본 논문의 구성은 다음과 같다. 2장에서는 Independence AAM을 간략하게 설명하고, 3장에서는 제안된 3D AAM알고리즘을 설명한다. 4장에서는 제안된 알고리즘의 성능을 평가하고 마지막 5장에서 결론을 맺는다.

2. Independence AAMs

AAMs는 Shape과 Appearance로 구성되며 이것의 목적은 모델이 입력영상에 최대한 잘 매치되도록 하는 파라미터를 Fitting Algorithm을 사용하여 찾아 새로운 모델 인스턴스를 생성하는 것이다.

AAMs의 Shape은 2D 영상에 n 개의 feature points를 표기하고, 각 점을 연결하여 생성된 삼각형 Mesh로 구성된다. 이것은 다음과 같이 벡터화하여 표현할 수 있다.

$$S = (x_1, y_1, x_2, y_2, \dots, x_n, y_n)^T \quad (1)$$

AAMs의 Shape은 선형변환으로 이루어 진다. 따라서 Shape S 는 Base Shape S_0 에 n 개의 Shape vectors S_i 의 선형조합을 더하는 것으로 다음과 같이 표현할 수 있다.

$$s = s_0 + \sum_{i=1}^n p_i s_i \quad (2)$$

식(2)에서 Shape vectors s_i 는 각 영상에 표기된 feature points의 좌표로 구성된 training set에 PCA(Principal Component Analysis)를 적용하여 계산한다. Base Shape s_0 는 Training Set의 평균이고 계수 p_i 는 Shape 파라미터이다.

AAMs의 Appearance는 각각의 영상에 있는 Shape내의 픽셀로 이루어 지고 이것은 Piecewise Affine Wapr[2]을 이용해서 추출할 수 있다. AAMs의 Appearance또한 선형변환으로 이루어 진다. 따라서 Shape s 와 마찬가지로 Appearance $A(x)$ 도 Appearance의 평균인 Base Appearance $A_0(x)$ 에 m 개의 Appearance vectors λ_i 의 선형조합을 더하는 것으로 다음과 같이 표현할 수 있다.

$$A(x) = A_0(x) + \sum_{i=1}^m \lambda_i A_i \quad (3)$$

식(3)에서 계수 λ_i 는 Appearance 파라미터 이다. 식(3)의 Base Appearance $A_0(x)$ 와 Appearance Vectors A_i 는 Shape과 마찬가지로 Training Images에 PCA를 적용하여 계산한다[1].

또한 AAMs는 2D Image Formation Model을 사용하는 데 이 모델은 $N(x; q)$ 로 표기되는 2D 닮음 변환(Similarity Transformation)을 사용한다. $N(x; q)$ 에서 $q = (q_1, q_2, q_3, q_4)^T$ 는 Rotation, Translation 그리고 Scale 파라미터를 담고 있다[2].

AAM Instance는 위에 설명한 Shape과 Appearance를 이용하여 생성한다. AAM Instance를 생성하기 위해서는 먼저 Shape의 p, q 파라미터를 계산하고 그것을 이용하여 AAMs의 Shape s 를 생성한다[2]. 그 다음 식(3)을 이용하여 AAM의 Appearance $A(x)$ 를 계산하는데 식(3)의 파라미터 λ_i 는 Shape 파라미터와 비슷한 방법으로 계산할 수 있다. 최종 Instance 는 위의 세 파라미터를 계산한 후 model shape s 와 Appearance $A(x)$ 를 갱신하여 Appearance $A(x)$ 를 model shape s 로 Warping하는 것이다. 따라서 AAM Instance는 다음과 같이 정의한다.

$$M(W(x; p; q)) = A(x) \quad (4)$$

식(4)에서 M 은 2D 영상을 나타내고 W 는 Warp을 뜻한다. 위 식에서 Warp은 Base Shape s_0 를 이루고 있는 Triangle과 대응하는 Model Shape s 에 Triangle간에 이루어진다.

3. 3D AAMs

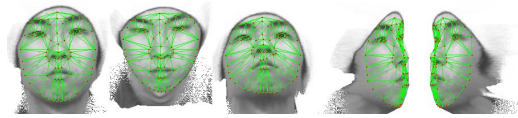
본 논문에서 제안한 3D AAMs은 3D Mesh와 2D 영상으

로 구성된 raw 3D data를 사용한다. 3D AAMs 구성은 raw 3D data의 3D Mesh와 2D 영상으로 2D AAMs와 Depth Models를 만들어 조합하는 방법을 사용한다.

3.1 Training set

본 논문에서 제안한 3D AAMs은 Raw 3D data를 사용해야 한다. 일반적으로 Raw 3D Data는 2D 영상과 3D Mesh로 이루어져 있다. 3D AAMs의 Shape Model은 3D를 표현해야 하기 때문에 Shape Model을 만들기 위한 Training Set은 3D 공간상의 Feature Points들이 삼각형으로 연결된 Mesh로 구성되어야 한다. 따라서 3D AAMs Feature Points는 Raw 3D Data가 Projection된 2D 영상에 손으로 표기한 Points와 대응하는 3D 공간상의 Points로 구성된다.

Depth를 가지고 있는 3D는 2D 와의 차이 때문에 하나의 2D영상을 사용해 3D 공간상의 Points를 추출하는 것은 적절하지 않은 Points를 추출할 수 있게 한다. 우리는 3D 공간상의 정확한 Feature Points를 추출하기 위하여 5개의 시점(정면, 좌/우 30°, 상/하 60°)에서 Projection된 2D 영상을 사용해 Feature Points를 추출한다. 그림(1)은 Raw 3D Data상에서 추출된 Feature Points를 연결한 Mesh를 보여준다.



정면 하 60° 상 60° 우 30° 좌 30°
그림 1. raw 3D data를 5개의 시점으로 Projection한 2D 영상.

3.2 Depth Model

3D AAMs의 Depth Model은 2D 영상의 밀도 차와 3D Mesh의 z축을 이용하여 생성한다. 일반적으로 일정한 환경에서 촬영한 2D 얼굴 영상에 밀도를 나타내는 픽셀 값들의 차는 입술이나 눈썹 등과 같은 신체적 특징 때문에 생기기도 하지만 주로 얼굴의 기하학적 특징에 의해 만들어진 음영 때문에 생기기도 한다. 따라서 이런 밀도 차는 얼굴의 기하학적 특징 정보를 반영하고 있다고 할 수 있다. 이런 밀도 차는 다음과 같은 식으로 구할 수 있다.

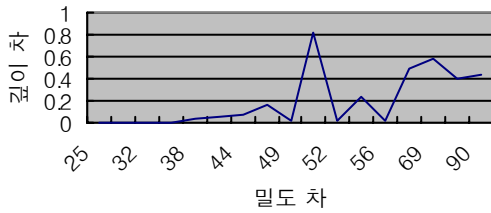
$$\bar{I} = [(I_{(x-1,y-1)} - I_{(x,y)}) + K + (I_{(x+1,y+1)} - I_{(x,y)})] / 8 \quad (5)$$

그림(2)는 식(5)를 이용해서 구한 개개의 밀도 차를 이미지화 한 것 이고 그래프(1)은 평균 밀도 차를 이미지화 한 것에 각 픽셀들을 참조하고 있는 3D Mesh들에 평균적인 z 축의 깊이 차를 보여주고 있다. 그래프(1)에서 깊이 차가 수 치적으로 1이하로 나오는 것은 깊이 비교를 특정한 한 픽셀을 둘러 싸고 있는 픽셀로 하기 때문에 픽셀을 참조하고 있는 Vertex가 근접해 있기 때문이다. 또한 그래프 상에서 가장 높은 밀도 차를 보이는 곳 보다 중간 정도의 밀도 차를 보이는 곳에 깊이 차가 더 큰 것은 입술 또는 눈썹과 같은 신체적 특징 때문이다. 비록 이런 이유로 밀도 차에 따른 깊

이 차가 비례하지 않다 하더라도 신체적 특징에 의한 밀도 차의 영역은 기하학적인 밀도 차의 영역과 비교하여 크기 않은 부분을 차지하기 때문에 AAM의 Shape을 생성하는데 큰 영향을 주지는 않는다.



그림 2. 2D 얼굴의 밀도 차.



그래프 1. 2D 얼굴의 밀도 차.

Depth Model 또한 선형 변환을 통해서 생산한다. 따라서 식(2)와 마찬가지로 다음과 같이 표현한다.

$$z = z_0 + \sum_{i=1}^n \beta_i c_i \quad (6)$$

위 식에서 β_i 는 모델 파라미터 이고 c_i 는 z축에 대한 벡터이다. c_i 는 AAM의 Shape과 Appearance와 마찬가지로 PCA를 적용해서 구한다.

3.3 Generating 3D AAM

3D AAMs의 Shape은 Raw 3D Data에서 추출한 n 개의 feature points에 x축과 y축을 사용하여 생성한 2D AAM Shape Model에 z축과 영상의 밀도 차를 이용해 생성한 Depth Model을 추가하는 것이다. 따라서 3D Shape \bar{s} 는 x, y축과 z축이 분리된 형태로 다음과 같이 정의할 수 있다.

$$\bar{s} = ((x_1, y_1, x_2, y_2, \dots, x_n, y_n), (z_1, z_2, \dots, z_n))^T \quad (7)$$

3D AAM에서 사용되는 Depth Model은 2D AAM의 Shape Model과 유사한 선형 변환을 한다. 따라서 3D AAM Shape은 2D AAM의 Shape과 Depth Model을 결합하는 것으로 표현할 수 있고 식(2)를 다음과 같이 확장하여 나타낼 수 있다.

$$\bar{s} = \left[s_0 + \sum_{i=0}^n p_i s_i \right] \left[z_0 + \sum_{i=0}^n \beta_i c_i \right] \quad (8)$$

3D AAMs의 Appearance $A(x)$ 는 Raw 3D data에 2D 영상을 사용 하고 생성은 2D AAM과 같은 방법을 사용한다. 3D AAMs Instance는 2D AAM의 파라미터 p, q, s_i 와 Depth Model 파라미터 β 의 결합이다. 따라서 식(4)에 확장으로 다음과 같이 정의한다.

$$M(P(W(x; p, q, \beta))) = A(x) \quad (9)$$

3DAAMs Instance는 2DAAMs Instance와 도 유사한 방법으로 생산되지만 Shape Model이 3D이기 때문에 3D Shape \bar{s} 를 2D frame으로 Projection하는 과정이 필요하다. 식(9)에서 P 는 고정된 Projection 파라미터를 의미하고, Projection은 [4]에서 사용한 Weak Perspective Projection으로 한다. 그림(3)은 3D AAMs Instance를 보여준다.



그림 3. 3D AAMs Instance

그림(2)에서 가장 좌측에 있는 영상은 입력 영상에 맞추어 AAM이 fitting을 한 것이고 우측 영상은 생성된 Model Instance를 각 시점 별로 투영한 것이다.

3.4 Fitting Algorithm

일반적으로 Fitting Algorithm은 입력 영상과 모델의 오차를 최소화 하여 입력 영상을 모델에 잘 매칭시키는 것을 말한다. 입력 영상을 모델에 매칭시키는 것은 Warp 파라미터를 통해서 이루어 지는데, 이러한 알고리즘은 [8]의 Lucas-Kanade Image Alignment에서 잘 설명되었고 다음과 같이 나타낼 수 있다.

$$\sum_x [A_0(x) - I(W(x; p))]^2 \quad (10)$$

식(12)에서 p 는 Warp 파라미터이며 이것은 모델과 입력 영상의 오차로 반복해서 갱신된다. p 의 갱신은 $p \leftarrow p + \Delta p$ 로 계산되는데 Δp 는 식(10)로 계산된 오차를 이용하여 현재의 파라미터 p 를 갱신하기 위해 계산된 파라미터 이다. Lucas-Kanade Image Alignment은 Forward Compositional Image Alignment에서 개선되었는데 이것은 다음의 식(11)과 같이 파라미터 p 와 Δp 를 조합하는 것으로 나타내어 진다.

$$\sum_x [A_0(x) - I(W(W(x; \Delta p); p))]^2 \quad (11)$$

이 알고리즘에 장점은 Lucase-Kanade Image Alignment 보다 fitting 속도가 빠르다는 것이다[6]. 최근에는 Forward Compositional Image Alignment를 개선한 ICIA(Inverse Compositional Image Alignment)를 가장 많이 사용하는데 이것은 식(12)와 같이 식(11)의 입력 영상과 모델의 역할을 역으로 놓은 것이다.

$$\sum_x [I(W(x; p)) - A_0(W(x; \Delta p))]^2 \quad (12)$$

식(12)의 파라미터 p 는 다음으로 구할 수 있다.

$$W(x; p) \leftarrow W(x; p) \circ W(x; \Delta p)^{-1} \quad (13)$$

파라미터 p 를 갱신하는 Δp 는 식(11)에 테일러 시리즈를 적용해 주어지는 식 :

$$\sum_{x \in s_0} \left[I(W(x; p)) - A_0(W(x; 0)) - \nabla A_0 \frac{\partial W}{\partial p} \Delta p \right]^2 \quad (14)$$

에 $W(x; 0)$ 를 Identity warp이라 가정하여 다음으로 계산할 수 있다.

$$\Delta p = H^{-1} \sum_x \left[\nabla A_0 \frac{\partial W}{\partial p} \right]^T [I(W(x; p)) - A_0(x)] \quad (15)$$

식(15)에서 $\frac{\partial W}{\partial p}$ 는 Warp Jacobian이고 ∇A_0 는 모델의 기울기 이미지이다. H 는 Hessian Matrix로서 다음과 같이 계산된다.

$$H = \sum_x \left[\nabla A_0 \frac{\partial W}{\partial p} \right]^T \left[\nabla A_0 \frac{\partial W}{\partial p} \right] \quad (16)$$

ICIA 알고리즘의 장점은 모델과 입력 영상의 역할을 역으로 놓기 때문에 Warp Jacobian $\frac{\partial W}{\partial p}$ 와 기울기 이미지 ∇A_0 를 고정된 모델에서 미리 한 번만 계산해 놓고 사용할 수 있기 때문에 앞의 두 알고리즘 보다 Fitting 속도가 빠르다는

것이다.

우리의 3D AAMs 알고리즘은 ICIA가 적용된 [2]에 real-time AAM fitting 알고리즘을 확장한 것이다. 따라서 식(17)로 표현되는 [2]에 알고리즘을 식(18)으로 확장해 표현한다.

$$\sum_{x \in s_0} \left[A_0(x) + \sum_{i=1}^n \lambda_i A_i(x) - I(W(x; p, q)) \right]^2 \quad (17)$$

$$\sum_{x \in s_0} \left[A_0(x) + \sum_{i=1}^n \lambda_i A_i(x) - I(P(W(x; p, q, \beta))) \right]^2 \quad (18)$$

앞서 우리는 3D AAM을 위해 2D AAM Shape과 Depth Model을 조합한다고 했다. 우리는 2D AAM Shape과 Depth Model을 조합하기 위해 2D AAM Shape 파라미터 p 와 닮음 변환 파라미터 q 그리고 Depth Model 파라미터 β 를 계산한다. Shape 파라미터 p 와 q 는 앞의 식 (12,13,14,15,16)와 [2]로 구할 수 있다.

Depth Model 파라미터 β 는 Shape 파라미터 p 와 유사하게 입력영상과 모델의 오차를 통해 갱신된다. β 의 갱신은 $\beta \leftarrow \beta + \Delta \beta$ 로 이루어 지고 $\Delta \beta$ 는 다음과 같이 계산된다.

$$\Delta \beta = H^{-1} \sum_x \left[\bar{A}_0 \frac{\partial z}{\partial c} \right]^T [I(W(x; p, q)) - A_0(x)] \quad (19)$$

위 식에서 \bar{A}_0 는 평균 얼굴 영상 $A_0(x)$ 에 식(5)를 적용

해서 획득한 밀도 차를 벡터화 한 것 이고 $\frac{\partial z}{\partial c}$ 는 z축에 고 유벡터 이다. 또한 Hessian Matrix H 는 식(18)과 유사하게 다음과 같이 계산한다.

$$H = \left[\bar{A}_0 \frac{\partial z}{\partial c} \right]^T \left[\bar{A}_0 \frac{\partial z}{\partial c} \right] \quad (20)$$

5. 실험 및 결과

우리는 제안된 알고리즘의 성능을 평가하기 위해 앞서 말한 stereo-camera based 3D face capturing device로 수집한 한국인 얼굴 데이터베이스를 기반으로 하는 얼굴 인식 시스템을 구축했다. 한국인 얼굴 데이터베이스는 3시즌 동안 110명으로부터 촬영한 218개의 3D 얼굴로 구성되어 있고 각각의 3D 얼굴은 정면에 무표정으로 제한되어 있다. 우리의 얼굴 평가 시스템은 3D AAMs 생성과 Gallery Set을 위해

시즌 1, 2의 3D 얼굴을 사용 하고 남은 시즌 3은 평가를 위한 Probe set으로 사용 했다. 다음 그림 4는 얼굴 인식 시스템 프로세스에 블록 다이어그램이다.

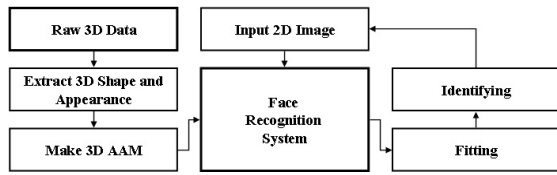


그림 4. Face Recognition System Process

비록 한국인 얼굴 데이터 베이스가 정면에 무표정으로 제한되어 있다 할지라도 2D영상은 투영 파라미터에 따라 변화될 수 있다. 우리는 얼굴 인식 시스템에 성능 평가를 다양하게 하기 위해서 Probe Set을 정면 뿐만 아니라 좌우 10~30°로 투영하여 성능을 테스트 하였다. Gallery Set과 Probe Set간의 얼굴 확인은 L2 + Mahalanobis distance 와 Angle + Mahalanobis distance를 사용하였다[11]. 얼굴인식 결과는 표(1)과 같다.

표 1. Recognition accuracy

	L2 + Maha	Angle + Maha
Front	91.6	83.3
Left 10°	91.9	80.6
Left 20°	83.6	75.4
Left 30°	56.3	59.6
Right 10°	90.3	79.0
Right 20°	85.2	72.1
Right 30°	70.8	66.6
Average	81.3	73.8

6. 결 론

본 논문에서 우리는 2D AAM과 Depth Model을 조합하는 새로운 3D AAM 알고리즘을 설명하였다. 또한 알고리즘을 평가하기 위해 stereo-camera로 촬영한 한국인 3D face database를 사용해 알고리즘을 구현한 후, 구현된 알고리즘으로 얼굴 인식을 수행하였다. 알고리즘 구현에는 100개의 3D face images에 3D Mesh와 2D 영상을 추출해 사용하였고 얼굴 인식에는 한국인 3D 얼굴 데이터베이스에 2D 영상만을 사용하였다. 얼굴 인식률은 앞서 본 것과 같이 높은 수치를 나타내었는데 이는 3D AAM Instance의 정확성 때문이라 할 수 있다. 하지만 본 알고리즘은 아직까지 Raw 3D Data의 많은 Vertices를 표현하지 못한다는 단점이 있다. 이를 위해 우리는 3D AAM의 Shape 뿐만이 아니라 Shape내에서도 Vertices를 생성해 보다 완벽한 3D를 표현할 수 있는 알고리즘을 연구하고 있다.

참고문헌

- [1] 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.
- [2] I. Matthews, S. Baker "Active Appearance Models revisited" *International Journal of Computer vision*, pages 135-164, 2004.
- [3] J. Xiao, S. Baker, I. Matthews, and T. Kanade. "Real-Time Combined 2D+3D Active Appearance Models" In *IEEE CVPR*, pages 535-542, 2004.
- [4] M. Jones and T. Poggio. "Multidimensional morphable models" In *Proceeding of the IEEE International Conference on Computer Vision*, pages 683-688, 1998.
- [5] S. Sclaroff and J. Isidoro. Active Blobs: "region-based, deformable appearance models" *Computer vision and Image Understanding*, pages 197-255, Feb. 2003.
- [6] S. Baker, I. Matthews "Lucas-Kanade 20 Years on: A Unifying Framework Part1", *International Journal of Computer Vision*, page 221-255, 2004.
- [7] A. Lanitis, C.J.Taylor, and T.F.Cootes. "Automatic interpretation and coding of face images using flexible models" *IEEE Transactions on Pattern Analyiss and Machine Intelligence*, pages 742-756, 1997.
- [8] 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.
- [9] T. H. Hong, H. B. Kim, H. J. Moon, Y. G. Kim, J. W. Lee, S. B. Moon. "Face Representation Method Using Pixel-to-Vertex Map(PVM) for 3D Model Based Face Recognition" *LNCS*, Vol.4031, pages 21-28, 2005
- [10] J. Xiao, J. Chai, and T.Kanade. "A closed-form solution to non-rigid shape and motion recovery" In *ECCV*, 2004.
- [11] H. Moon and P. Phillips. Computational and Performance Aspects of PCA-Based Face Recognition Algorithms. *Perception*. Vol. 30 pages 303-321, 2001.
- [12] M. Brand. Morphable 3D models from video. In *Proceedings of CVPR*, 2001.
- [13] S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3D morphable model. In *ICCV*, 2003
- [14] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *proceedings of CVPR*, 200.

- [15] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, vol. 3, pages. 71-86, 1991.