

Tensor 기반의 Multi-linear Analysis 를 이용한 Active Appearance Model

Active Appearance Model using Multi-linear Analysis based on Tensor

조경식, Gyeong-Sic Jo*, 김용국, Yong-Guk Kim**

요약 Active Appearance Models(AAMs)은 얼굴인식, 얼굴추적, 표정인식 뿐만 아니라 눈동자 추적과 같은 분야에도 적용되어 좋은 성능을 보여 주었다. 보통 AAM 을 생성하기 위해서는 얼굴 영상과 얼굴의 특징을 나타내는 점으로 구성된 매쉬로 이루어 지는 트레이닝 셋이 필요하다. AAM fitting algorithm 은 학습한 얼굴과 유사한 얼굴을 Fitting 할 때에는 뛰어난 성능을 보이지만 조명에 의한 그림자 또는 액세서리에 의한 얼굴의 피부 가림과 같이 전체 얼굴이 잘 나타나지 않는 불완전한 영상의 Fitting 은 입력영상과 템플릿 영상간의 오차가 커지기 때문에 실패할 가능성이 매우 높다. 본 논문에서 우리는 AAMs 에서 사용되는 PCA 를 Higher-order Singular Value Decomposition(HOSVD)로 대체하여 이 문제를 보완하는 강화된 AAM 을 제안한다. 제안된 AAM 에는 기존에 사용하던 고유벡터와 함께 HOSVD 를 통해 획득할 수 있는 Eigen-Modes 를 추가하여 사용한다. 또한 우리는 Yale Face Database 를 이용한 평가를 통해 제안된 AAM 이 기존 AAM 보다 불완전한 영상에 효과적으로 대응하는 것을 보여준다.

핵심어: AAM, PCA, HOSVD, Yale Face Database

*주저자 : 세종 대학교 컴퓨터 공학과 e-mail: nopnse@sju.ac.kr

**공동저자 : 세종 대학교 컴퓨터 공학과 교수 e-mail: ykim@sejong.ac.kr

1. 서론

Active Appearance Models(AAMs)는 파라메트릭 모델로서 얼굴 모델링에 주로 사용되고 있으나 다른 객체

모델링에도 잘 적용될 수 있다. 또한 AAM의 다양한 실시간 영상 어플리케이션은 다이내믹 포즈 추론, 표정인식 그리고 립-리딩과 같은 것에도 사용될 수 있다[4]. AAM은 입력 영상이 [4]에서 설명한 것과 같이 얼굴의 3D pose variation, 조명, 헤어, 그리고 악세사리에 의한 불완전한 얼굴을 담고 있다면 fitting error가 커지기 때문에 fitting이 정확히 되지 않는다는 문제점을 가지고 있다. 우리는 이를 해결하기 위해 [2]의 AAMs fitting algorithm에 Principal Components Analysis(PCA)에서 획득한 벡터가 아닌 [6]에서 설명된 Higher-order Singular Value Decomposition(HOSVD)에서 획득한 벡터를 적용하여 불완전 영상에서도 좋은 fitting 결과를 얻을 수 있도록 하였다. Traditional PCA는 전체 데이터 셋의 subspace만을 만들지만 HOSVD는 Singular Value Decomposition 또는 PCA의 확장으로써 Tensor를 기반으로 하여 각 order의 subspace를 만들 수 있고 이를 이용하여 각 모드의 주성분을 자세히 표현할 수 있는 Eigen-Modes를 생성할 수 있다. 따라서 같은 데이터 셋으로 트레이닝을 하더라도 HOSVD의 각 subspace를 이용하면 PCA보다 데이터를 더 잘 컨트롤할 수 있다. 우리는 HOSVD의 subspaces를 이용하여 각 모드의 Subspace를 생성하고 이를 이용하여 Eigen-Modes를 계산하여 이를 AAM의 Image Alignment Algorithm에 적용하여 제안된 AAM이 기존의 AAM보다 성능이 향상되었다는 것을 보여준다.

2. Active Appearance Model

[2]의 AAM은 shape과 appearance를 나누어 정의한다.

2.1 Shape and Appearance

AAM의 Shape은 트레이닝 셋의 메쉬를 이루는 버텍스의 좌표로 정의된다. AAM은 선형 변환을 하므로 Shape 모델은 다음과 같이 표현된다.

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

여기서 s_0 는 평균 shape이고 s_i 는 shape vectors이다 그리고 p_i 는 shape모델의 파라미터를 의미한다. shape vectors s_i 는 트레이닝 셋의 shape 좌표에 Procrustes analysis를 한 후 PCA를 이용하면 얻을 수 있다. AAM의 Appearance 모델은 평균 shape s_0 내에서 정의된다[2]. 이 뜻은 각 영상의 Shape안쪽에 있는 픽셀을 Shape s_0 로 놓아야 한다는 것이다. Appearance 모델 또한 선형 변환을 한다. 따라서 Appearance 모델도 식(1)과 유사하게 표현된다.

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

식(2)에서 A_0 는 평균 Appearance이고 A_i 는 Appearance vectors이다 그리고 λ_i 는 Appearance모델의 파라미터를 의미한다.

AAM shape 파라미터 p_i 와 Appearance 파라미터 λ_i 가 주어지면 AAM 모델 인스턴스를 생성할 수 있다. 모델 인스턴스는 파라미터 p_i 로 생성한 최신 shape에 평균 shape s_0 내에서 생성된 최신 Appearance를 warping하는 것으로 정의되고 이것은 다음과 같은 식으로 표현할 수 있다.

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

여기서 M 은 appearance가 shape으로 warping된 이미지를 뜻한다.

3. Multi-linear Analysis

3.1 Tensor Algebra

Multi-linear algebra는 higher-order tensors를 기반으로 한다. Tensor는 Multi-linear algebra 또는 n -mode matrix로 잘 알려져 있으며 벡터와 행렬을 일반화한 것이다. 따라서 벡터는 first-order tensor이고 행렬은 second-order tensor가 되며, 이러한 Multi-linear Analysis는 다양한 컴퓨터 비전과 그래픽스 문제에 잘 접근할 수 있는 통일된 수학적 프레임 워크를 제공한다[7].

M th order tensor는 $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ 로 주어지고 N 는 Tensor의 order를 나타낸다. Tensor \mathcal{A} 는 컬럼 벡터를 누적하는 것으로 $A_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times I_2 \times \dots \times I_{n-1} \times I_{n+1} \times \dots \times I_N)}$ 와 같이 행렬로 나타내는 unfolding이 가능하고, 이것은 Tensor를 간단한 행렬 연산으로 다룰 수 있도록 해준다.

Tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ 와 행렬 $M \in \mathbb{R}^{I_n \times J_n}$ 의 mode- n 곱은 $B = \mathcal{A} \times_n M$ 으로 표기되고 $B \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_{n-1} \times J_n \times I_{n+1} \times \dots \times I_N}$ 의 항목은

$$(\mathcal{A} \times_n M)_{i_1 \times \dots \times i_{n-1} \times j_n \times i_{n+1} \times \dots \times i_N} = \sum_{i_n} a_{i_1 \times \dots \times i_{n-1} \times j_n \times i_{n+1} \times \dots \times i_N} m_{j_n i_n} \quad (4)$$

으로 계산된다. mode- n product를 tensor로 표기하면 다음과 같이 나타낼 수 있고 :

$$B = \mathcal{A} \times_n M \quad (5)$$

식(7)은 다음과 같이 unfolded 행렬로도 표현할 수 있다.

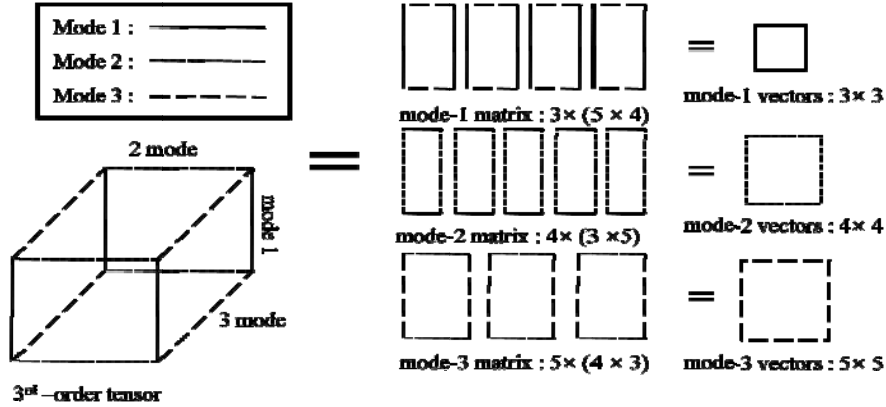


그림 1. Unfolding a 3rd order tensor of dimensions 3×4×5 and mode-*n* vectors.

$$B_{(n)} = MA_{(n)} \quad (6)$$

3.2 Decomposition

Order *N* tensor 는 *N* spaces 로 이루어 지며 HOSVD 를 적용하면 *N*-orthogonal spaces 와 core tensor 로 Decomposing 할 수 있다.

$$\mathcal{D} = \mathcal{Z} \times_1 U_1 \times_2 \dots \times_n U_n \dots \times_N U_N \quad (7)$$

식(7)에서 U_n 는 각 mode의 subspaces를 의미하며 Tensor \mathcal{D} 를 Unfolding한 행렬 $D_{(n)}$ 의 컬럼을 spanning 하는 orthonormal vectors로 구성되며, 각 subspaces에 column vector는 다른 모드 행렬에 영향을 준다. 각 Subspaces를 계산하기 위해서는 Order *N* tensor를 Unfolding 해야 하며 그림 1은 3rd - order tensor의 Unfolding 과정을 보여준다. 식(7)에서 \mathcal{Z} 는 mode matrices U_n 간의 Interaction을 관리하는 core tensor를 의미하며 SVD의 diagonal singular value matrix와 유사 하지만 diagonal 구조는 아니다. HOSVD를 이용한 Decomposing 알고리즘은 다음과 같은 과정으로 요약될 수 있다.

- i. Unfolded 행렬 $D_{(N)}$ 를 SVD로 계산하고 왼쪽 행렬로 U_N 를 선택한다.
- ii. Core tensor를 계산한다.

$$\mathcal{Z} = \mathcal{D} \times_1 U_1^T \times_2 \dots \times_n U_n^T \dots \times_N U_N^T \quad (8)$$

식(8)은 다음과 같이 행렬로도 계산할 수 있다.

$$\mathcal{Z}_n = U_N^T \times D_N \times (U_1 \otimes U_2 \otimes \dots \otimes U_{n-1}) \quad (9)$$

위 식에서 \otimes 는 kronecker product를 의미한다. 식(9)은 각 subspace에 cut off를 적용하였을 때 유용하게 사용될

수 있다.

4. Applying Multi-Linear Analysis to AAM

Multi-linear analysis 의 장점은 Core Tensor \mathcal{Z} 가 Subspace 를 각 모드에 주성분을 혼합적으로 표현하는 Eigen-Modes 로 변환하는 것이 가능하다는 것이다. Eigen-modes 는 $\mathcal{Z} \times_N U_N$ 로 쉽게 계산할 수 있다.

우리는 AAM 에 Multi-Linear Analysis 를 적용하기 위해서 third-order tensor $\mathcal{D} \in \mathbb{R}^{I \times J \times K}$ 를 구성한다. 구성된 Tensor \mathcal{D} 에 *I*, *J* 그리고 *K* 는 각각 사람, 포즈, feature 를 나타낸다. 우리는 Tensor \mathcal{D} 를 Decomposing 하기 위하여 다음과 같이 HOSVD 를 사용한다.

$$\mathcal{D} = \mathcal{Z} \times_1 U_{id} \times_2 U_p \times_3 U_f \quad (10)$$

위 식에서 U_{id} , U_p , 그리고 U_f 는 사람의 subspace, pose 의 subspace 그리고 feature 의 subspace 를 나타낸다. 우리는 식(10)의 Decomposing 을 이용하여 AAM 에 적용될 Eigen-Modes 를 다음과 같이 생성한다.

$$\mathcal{O} = \mathcal{Z} \times_3 U_f \quad (11)$$

위의 식(10)과 (11)를 이용해 계산된 Eigen-Modes 는 각 모드의 Subspace 와 Tensor \mathcal{D} 가 완전한 상태에서만 정상적으로 계산된다는 단점이 있다. 예를 들어 전통적인 PCA 에서 계산된 Subspace 에 Cut off 를 적용하는 것과 같이 각 모드의 Subspace 에 Cut off 가 적용되면 식(10)과 (11)을 계산하는 것이 어렵게 된다. 이를 해결하기 위하여 우리는 식(9)의 행렬연산을 사용한다.

$$\mathcal{O} = U_f \times (U_N^T \times D_N \times (U_1 \otimes U_2 \otimes \dots \otimes U_{n-1})) \quad (12)$$

식(12)의 장점은 Subspace 에 Cut Off 를 적용하더라도 Eigen-Modes 를 계산할 수 있어 원하는 높은 고유 값에 대응하는 Vectors 만을 사용해 Eigen-modes 를 구할 수 있다는 것이다.

AAM 에 mode vectors 가 적용됨에 따라 식(1)은 다음과 같이 새로이 표현 된다.

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^n p_i \mathbf{s}_i + \sum_{j=1}^m r_j \mathbf{o}_j \quad (9)$$

위 식에서 \mathbf{o}_j 는 mode vectors 이고 r_j 은 mode vectors 의 파라미터이다.

Eigen-modes \mathbf{O} 가 적용된 AAM 의 장점은 AAM Fitting Algorithm 인 Inverse Compositional Image Alignment(ICIA)에 쉽게 적용될 수 있어 [2]의 AAM 에 강점인 빠른 fitting 속도를 그대로 유지할 수 있어 실시간 영상에서도 잘 적용될 수 있고, 또한 Eigen-modes 를 높은 고유 값에 대응하는 고유 벡터들로 생성하여 입력 영상과 템플릿 영상간의 오차가 크더라도 AAM shape 이 오차에 강건하게 대응할 수 있도록 한다는 것이다. 그림(2)는 제안된 AAM 과 기존의 AAM 이 포즈와 조명 변화 때문에 발생하는 템플릿 영상과의 큰 오차 하에서 Fitting 을 수행한 결과를 보여준다.

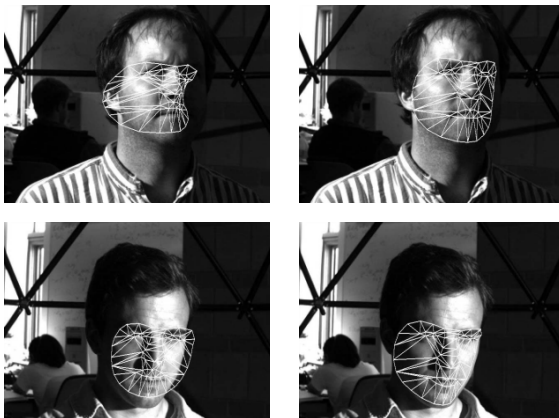


그림 2. 왼쪽 영상은 기존 AAM의 Fitting 결과이고, 오른쪽은 제안된 AAM의 Fitting 결과 이다.

5. 결론

우리는 AAM 의 생성과 평가에 Yale face database B[8]를 사용하였고 DB 에는 9 Poses 에 각각 64 Illuminations 이 적용된 10 명의 Images 를 가지고 있다. Yale Database 는 사람과 자세가 같은 카테고리에 포함되어 있다면 조명에 변화가 있더라도 얼굴의 위치는 같은 좌표상에 있게 된다. 다시 말해 영상에 조명의 변화가 있을 지라도 영상에 표기해 놓은 Landmarks 의

좌표는 같은 카테고리의 모든 영상에 얼굴을 나타낼 수 있다는 것이다. 우리는 이런 Yale Database 의 특징을 이용하여 test set 과 train set 을 구성하였다.

AAM 과 Multi-linear Analysis 가 적용된 AAM(MAAM)을 생성하기 위한 Train set 은 9 명의 9poses 에서 Illumination 상태가 좋은 1 개의 영상을 선택하여 구성하였으며 각 영상에는 64 개의 Landmarks 가 표기 되어 있다. Test set 은 Train set 과 사람과 자세가 같지만 조명이 다른 영상들로 구성되었으며, Ground Truth 는 Train set 의 각 영상에 표기된 Landmarks 로 만들어 진다. 평가는 두 AAM 의 fitting 속도와 Shape 이 Landmarks 내의 얼굴을 얼마나 정확하게 찾는지를 비교하는 것으로 수행 된다.

5.1 Efficiency Comparison

제안된 AAM 은 Original AAM 보다 많은 parameter 를 사용하기 때문에 Face Tracking 과 같은 실시간 영상에 적용하기 위해서는 기존의 AAM 과 같이 빠른 Fitting 속도를 보여야 한다. 우리는 Quad CPU 2.4GHz 에 2GB RAM 을 탑재한 PC 에서 Test 영상에 총 5 회의 Fitting Algorithm 을 수행하고 각각에 소요 시간과 평균 소요 시간을 측정 하였다. 두 AAM 에는 같은 수의 shape parameter 가 적용 되었고 제안된 AAM 에는 mode parameter 가 추가 되었으며 두 AAM 의 Appearance 는 7594 개의 픽셀을 담고 있다. 표 1 은 제안된 AAM 과 기존 AAM 의 속도를 비교한 것으로 둘의 속도 차이가 거의 없다는 것을 보여준다.

	1	2	3	4	5	Avg.
Traditional AAM	7ms	6ms	6ms	6ms	7ms	6.4ms
Suggested AAM	7ms	7ms	7ms	7ms	7ms	7ms

5.2 Robustness Experiments

우리는 제안된 AAM 이 Template $T(x)$ 와 입력 영상 $I(x)$ 사이에서 발생하는 큰 오차에서도 Shape 을 얼굴에 얼마나 강건하고 안정적으로 Fitting 할 수 있는 지를 실험한다. 실험은 AAM 의 Template $T(x)$ 와 입력영상 $I(x)$ 의 Mesh 내의 픽셀에 오차를 계산하여 분류한 후 각 파트에 Fitting Algorithm 을 적용하여 Algorithm 에 Iteration 별 Shape 오차를 계산하는 것으로 수행되었다. Shape 의 오차는 Ground Truth 와 두 AAM 의 Shape 에 오차를 City Block Distance 로 계산 한다. 우리는 실험을 위해 영상의 조명과 포즈에 따라 나타나는 Test set 의 픽셀당 오차율을 계산하여 4 개의 Set 으로 분류하였고 각 set 에는 평균 오차율이 40~49, 50~59, 60~69, 그리고 70~79 인 영상들이 담겨 있다. 그림(3)은 제안된 AAM 과 기존 AAM 의 실험결과를 보여준다. 그림(3)에서 Graph 는 각 set 의 Iteration 별 평균 Shape 오차를 나타낸다.

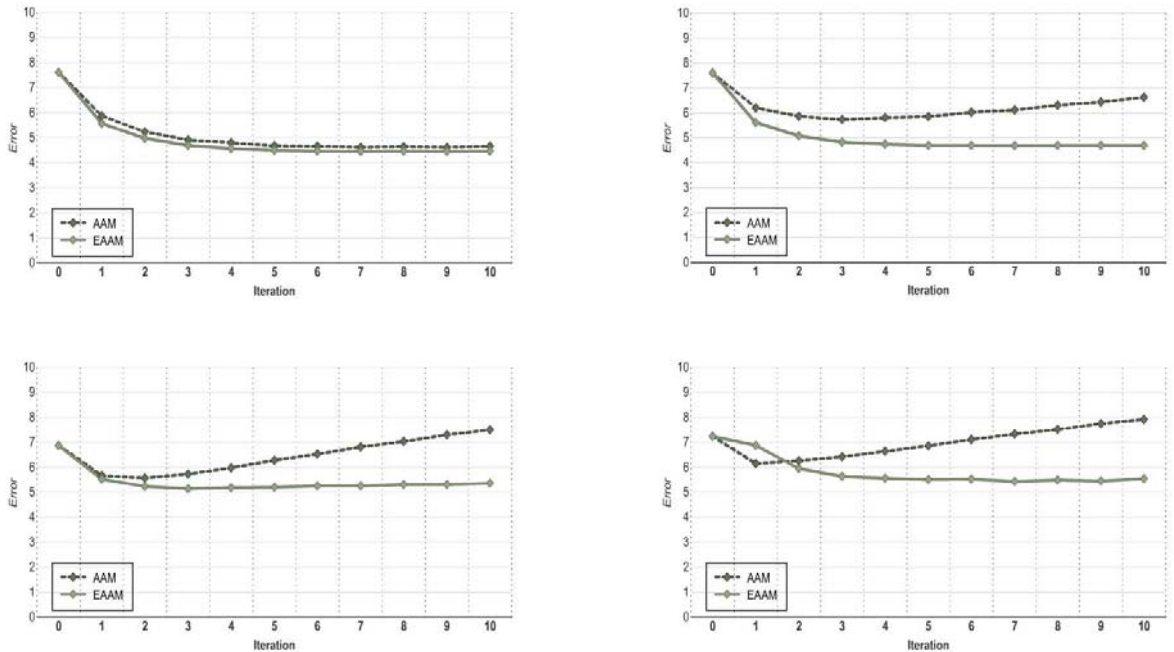


그림 3. Template $T(x)$ 와 Input image $I(x)$ 의 오차와 두 AAM의 Fitting이 수행된 후의 shape과 Ground Truth의 좌표 오차. 각 Graph는 픽셀당 평균 오차가 40~49(Top left), 50~59(Top right), 60~69(Bottom left), 그리고 70~79(Bottom right)인 영상들에 Shape 오차를 보여준다.

Graph에서 기존 AAM의 Shape은 $T(x)$ 와 $I(x)$ 의 픽셀 값 오차가 증가할수록 Shape 오차 또한 이와 비례하여 증가하고, 제안된 AAM의 Shape은 오차가 증가할지라도 Shape의 오차가 크게 증가하지 않는다는 것을 보여준다. 이것은 기존의 AAM은 Shape이 얼굴 영역에 위치하고 있을지라도 음영이 발생한 곳은 $T(x)$ 와 $I(x)$ 의 픽셀 값 오차크기 때문에 Shape을 픽셀 값 오차가 적은 쪽으로 Fitting하고, 제안된 AAM은 $T(x)$ 와 $I(x)$ 의 픽셀 값 오차가 크더라도 Eigen-modes 0의 영향으로 Shape을 안정적으로 유지하며 fitting을 한다는 것을 보여준다. 이런 Fitting 결과는 그림(3)에서도 확인 할 수 있다.

6. Conclusion

본 논문에서 우리는 Eigen-modes가 적용된 AAM을 제안하였다. Eigen-modes가 적용된 AAM을 만들기 위해 우리는 Tensor에 HOSVD를 적용해 Decomposing을 하여 3개의 subspace 만든 후 Eigen-modes를 생성하였고 이를 AAM의 Image Alignment Algorithm인 ICIA에 적용하였다. 실험에서 제안된 AAM은 Eigen-modes가 추가되어 파라미터가 증가했음에도 ICIA의 빠른 Fitting 속도를 유지함으로써 실시간 영상 처리에도 잘 적용될 수 있음과, 조명과 포즈로 인해 생기는 폐색에도 강건하게 대처할 수 있다는 것을 보여주었다. 비록 제안된 AAM이 still 영상에 성공적으로 적용되었고 실시간 영상에 적용될 만큼 빠른 Fitting 속도를 보유했다는 것을 보여 주었지만 아직까지

실시간 영상에는 적용되지 않았다. 따라서 우리는 제안된 AAM을 실시간 영상에 적용하고 이를 확장하여 실시간 표정인식과 같은 분야에도 적용할 계획이다.

7. Acknowledgements

본 연구는 서울시 클러스터 프로젝트의 지원을 받았습니다.

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, June 1998, pp. 300-3005.
- [2] I. Matthews, S. Baker "Active Appearance Models revisited" International Journal of Computer vision, 2004, pp 135-164.
- [3] T. Cootes, G. Edwards, and C. Taylor. "Active appearance models", IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(6), 2001.
- [4] R. Gross, I. Matthews, and S. Baker. "Constructing and fitting active appearance models

with occlusion” , In proceedings of the IEEE Workshop on face processing in Video, Jun 2004.

[5] M. Turk and A. Pentland, “Eigenfaces for Recognition” , Journal of Cognitive Neuroscienc, vol. 3, no.1, 1991, pp. 71–86.

[6] L. De Lathauwer, B. De Moor, and J. Vandewalle, “A Multilinear Singular Value Decomposition” , SIAM Journal of Matrix Analysis and Applications, 21(4), 2000.

[7] M. A. O. Vasilescu and D. Terzopoulos, “Multilinear analysis of image ensembles : Tensorfaces” , In 7th European Conference on Computer Vision, 2002, pp. 447–460.

[8] A. S. Georghiades, P. N. Belhumeur, and K. J. Kriegman, “From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose” , IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no. 6, 2001, pp. 643–660.