

# KPCA 기반 노이즈 제거 기법을 이용한 부분 손상된 얼굴 영상의 복원

## Reconstruction of Partially Occluded Facial Image Utilizing KPCA-based Denoising Method

강대성, 김종호, 박주영  
고려대학교 제어계측공학과

Daesung Kang, Jongho Kim, Jooyoung Park  
Dept. of Control and Instrumentation Engineering, Korea University  
E-mail : {mpkds, oyeasw, jpark}@korea.ac.kr

### 요약

많은 경우, 부분 손상된 얼굴 영상을 복원해야 할 필요가 있다. 대표적인 예로는 감시 카메라에 찍힌 범인의 얼굴 영상이 이에 속한다. 이런 경우 얼굴의 중요한 부분이 가려져 있기 때문에 자동 얼굴 인식 시스템이나 사람의 관찰로는 그 부분을 인식하기는 매우 어렵다. 이 논문에서는 그 어려움을 극복하기 위해 Kernel PCA 기반 노이즈 제거 기법을 부분 손상된 얼굴 영상에 적용한 문제를 고려해 보았다.

### Abstract

In numerous occasions, there is need to reconstruct partially occluded facial image. Typical examples include the recognition of criminals whose facial images are captured by surveillance cameras. In such cases a significant part of the face is occluded making the process of identification extremely difficult, both for automatic face recognition systems and human observers. To overcome these difficulties, we consider the application of Kernel PCA-based denoising method to partially occluded facial image in this paper.

키워드 : Occluded facial image, KPCA (Kernel Principal Components Analysis)

### 1. 서론

얼굴 인식에 대한 많은 연구가 오랫동안 계속되고 있고, 좋은 결과가 많이 나왔다. 하지만 아직까지 부분 손상된 얼굴을 인식하는 데에는 어려움이 많다. 예를 들면, 폐쇄회로 TV(CC-TV) 화면 등을 통해 얼굴이 드러났더라도 모자, 선글라스, 복면 등으로 얼굴의 일부만 노출될 경우 신원 확인이 어렵다. 이 어려움을 해결하기 위해서 본 논문에서는 최근에 많은 연구가 진행되고 있는 커널 주성분 분석 기법(Schölkopf & Smola, 2002 [2]; Vapnik, 1998 [6])을 손상된 얼굴 영

상을 복원하는데 적용하였다. 기존의 주성분 분석 기법(PCA)은 입력 공간(input space)  $X$ 에 데이터들이 선형적으로 분포되어 있을 경우에는 좋은 결과를 얻을 수 있었지만, 데이터들이 비선형적으로 분포되어 있는 경우에는 좋은 결과를 얻을 수 없었다. 반면 커널 주성분 분석 기법은 입력 공간  $X$ 에 비선형적으로 분포해있는 데이터를 비선형 매핑 연산자  $\varphi$ 를 통해서 특징 공간(feature space)으로 매핑하게 되면 데이터들이 선형적으로 분포하게 되는데, 이 데이터들을 특징 공간에서 주성분 분석 기법(PCA)을 사용하는 방법론이다. 특히 본 논문은 비선형 최적화에 의존하는 전통적인 기법(Mika et al., 1998 [3])이

아니라 입력 공간에서의 두 점과 특징 공간에서의 두 점 사이의 거리 간에 성립하는 간단한 관계식(Williams, 2001 [5])을 이용해서 노이즈가 제거된 pre-image를 직접적으로 찾는 기법(J.T. Kwok et al., 2004 [4])을 이용하여 얼굴 영상을 복원을 시도했다.

본 논문의 구성은 다음과 같다. 2장에서는 커널 주성분 분석 기법에 대해서 간단히 기술하고, 3장에서는 시뮬레이션 결과를 보이고 분석한다. 마지막으로 4장에서는 결론과 향후과제에 대해서 논한다.

## 2. Kernel PCA 기법

### 2.1 특징 공간에서의 PCA

Kernel PCA 기법은 PCA 기법과 매우 유사하다. 다만 다른 점이 있다면 Kernel PCA 기법은 특징 공간에서 주성분 분석을 한다는 것이다. 특징 공간에서의 PCA는 다음과 같은 식으로 하면 된다. 얼굴 영상의 패턴 집합  $\{x_1, x_2, \dots, x_N\}$  의 커널 행렬  $K = [K_{ij}] = k(x_i, x_j)$  을 먼저 구한다. 그리고 커널 행렬의 평균값을 기준으로 평행이동시킨다. 그리고 그 행렬의 고유값(eigenvalue)  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_N)$  과 고유벡터(eigenvector)  $U = [\alpha_1, \dots, \alpha_N]$  를 각각 구한다. 여기서 구한 고유값과 고유벡터를 이용해서 특징 공간에서  $k$  번째 orthonormal 고유벡터를 구하는데 그 식은 다음과 같다.

$$V_k = \sum_{i=1}^N \frac{\alpha_{ki}}{\sqrt{\lambda_k}} \varphi(x_i)$$

그리고 테스트 벡터  $\varphi(x)$ 의  $k$  번째 특징 값  $\beta_k$  는  $\varphi(x)$ 와  $V_k$ 의 내적으로 구해지는데 그 식은 다음과 같이 표현된다.

$$\beta_k = \varphi(x)^T V_k$$

$\varphi(x)$ 를 고유벡터에 투영시킨  $P\varphi(x)$ 는 벡터 성분  $\beta_k$ 와 정규직교 고유벡터  $V_k$ 의 선형 결합으로 이루어진다.

$$P\varphi(x) = \sum_{k=1}^K \beta_k V_k$$

### 2.2 특징 공간과 입력 공간에서의 거리

어떤 두 패턴  $x$ 와  $x_i$ 에 대해서, 특징 공간에서의 두 점  $P\varphi(x)$ 와  $\varphi(x_i)$ 사이의 거리의 제곱은 다음과 같다.

$$\begin{aligned} \tilde{d}^2(P\varphi(x), \varphi(x_i)) &= \|P\varphi(x)\|^2 + \|\varphi(x_i)\|^2 \\ &\quad - 2P\varphi(x)^T \varphi(x_i) \end{aligned}$$

그리고 입력 공간에서의 두 점 사이의 거리  $d_{ij}$ 는 쉽게 구할 수 있다. 입력 공간에서의 거리  $d_{ij}$ 와 특징 공간에서의 거리  $\tilde{d}_{ij}$  사이에는 다음과 같은 간단한 관계식이 성립한다.

$$\begin{aligned} \tilde{d}_{ij}^2 = \tilde{d}^2(x_i, x_j) &= K_{ii} + K_{jj} - 2k(\|x_i - x_j\|^2) \\ &= K_{ii} + K_{jj} - 2k(d_{ij}^2) \end{aligned}$$

(여기서  $K_{ii} = K_{jj} = 1$  이 된다.)

결론적으로  $k(d_{ij}^2) = \frac{1}{2}(K_{ii} + K_{jj} - \tilde{d}_{ij}^2)$  이 된다.

보통  $k$ 는 역변환이 가능하다. 가우스 커널의 경우  $k(z) = \exp(-\beta z)$ 가 되고,  $\beta$ 는 상수이다. 여기서 입력 공간에서의 거리를 대수적으로 구할 수 있게 되고 그 식은 다음과 같이 얻을 수 있다.

$$d_{ij}^2 = -\frac{1}{\beta} \log\left(\frac{1}{2}(K_{ii} + K_{jj} - \tilde{d}_{ij}^2)\right) \quad (1)$$

### 2.3 거리 제한에 기초를 둔 pre-image 구하기

2.2절을 통해서 특징 공간에서의  $P\varphi(x)$ 와 각 학습 패턴  $x_i$ 간의 거리인  $\tilde{d}^2(P\varphi(x), \varphi(x_i))$ 를 알기 때문에 쉽게 입력 공간에서의  $P\varphi(x)$ 의 pre-image  $\hat{x}$ 와 각 학습 패턴  $x_i$ 사이의 거리를 구할 수 있다. 보통 어떤 점의 위치를 결정하는데 가장 큰 역할을 하는 것이 이웃에 있는 점들간의 거리이다. 그래서 본 논문에서는  $x$ 에 이웃하는  $n$ 개의 가장 가까운 패턴 벡터  $\{x_1, x_2, \dots, x_n\} \in R^d$ 를 거리 계산할 때 고려하였다.  $x$ 와 이웃하는 벡터간의 거리는 다음과 같이 정의된다.

$$d^2 = [d_1^2, d_2^2, \dots, d_n^2]^T$$

본 논문에서는  $P\varphi(x)$ 를 입력 공간으로 매핑시키기 위해서 MDS [7]개념을 사용한다. 이를 위해 먼저 학습 데이터  $X = [x_1, x_2, \dots, x_n]$ 의 중심

심  $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ 을 얻기 위해서 평균을 취한 후,

$d \times n$  행렬  $X = [x_1, x_2, \dots, x_n]$ 을 만든다. 그리고  $n \times n$  centering 행렬을 정의하면 다음 식과 같이 정의된다.

$$H = I_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T$$

여기서  $I_n = \text{diag}[1, \dots, 1] \in R^{n \times n}$ 인 행렬이고  $\mathbf{1}_n = [1, \dots, 1]^T \in R^{n \times 1}$ 인 벡터이다. 행렬  $XH$ 는  $x_i$  벡터들을 그들의 중심으로 평행이동시킨다. 즉  $XH = [x_1 - \bar{x}, \dots, x_n - \bar{x}]$ 이 된다. 그리고  $XH$ 의 열 공간 내에서 좌표계를 정의한다. 만약  $XH$ 의 rank가  $q$ 라면, 우리는  $d \times n$ 행렬  $XH$

의 SVD (singular value decomposition) [8]을 다음과 같이 구할 수 있다.

$$XH = [U_1 \ U_2] \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$$

$$= U_1 \Lambda V_1^T = UZ$$

$U_1 = [e_1, \dots, e_q]$  은 orthonormal 열벡터  $e_i$ 로 이루어진  $d \times q$  행렬이고,  $Z = \Lambda V_1^T = [z_1, \dots, z_n]$  는  $e_j$  위로  $x_i - \bar{x}$ 을 투영한 열벡터  $z_i$ 로 이루어진  $q \times n$  행렬이다.  $x_i$ 와  $z_i$ 는 다음과 같은 식을 만족한다.

$$\|x_i - \bar{x}\|^2 = \|z_i\|^2, \quad i=1, \dots, n$$

이를  $n$ 차 벡터까지 집합시키면 다음과 같다.

$$d_0^2 = [\|z_1\|^2, \dots, \|z_n\|^2]^T$$

앞에서 언급한대로 pre-image  $\hat{x}$ 의 위치는  $d^2(\hat{x}, x_i)$ 를 계산해서 얻을 수 있다.

$$d^2(\hat{x}, x_i) \simeq d_i^2, \quad i=1, \dots, n$$

이상적인 경우, 이들 간의 거리는 반드시 일정해야 한다. 하지만, 대개 입력 공간에서 정확한 pre-image가 존재하지 않고, 심지어는 이들 간의 거리조건을 만족하는 해답이 존재하지 않을 수도 있다. 그렇기 때문에  $\hat{x}$ 를 구하기 위해서는 최소자승법(Least Squares)을 이용해야 한다. 최소자승법을 이용해서  $\hat{z}$ 를 구하면 다음과 같다.

$$\hat{z} = -\frac{1}{2} \Lambda^{-1} V_1^T (d^2 - d_0^2)$$

$\hat{z}$ 와  $\hat{x}$ 는 다음과 같은 관계를 가지고 있다.

$$\hat{x} - \bar{x} = U_1 \hat{z}$$

결국 노이즈가 제거된 패턴의 위치는 다음과 같이 구할 수 있다.

$$\hat{x} = U_1 \hat{z} + \bar{x}$$

### 3. 시뮬레이션 결과 및 분석

#### 3.1 얼굴 데이터베이스

이번 시뮬레이션에 사용한 얼굴 영상<sup>1)</sup>들은 모두 200개의 2차원 백인 얼굴 이미지를 사용하였는데 이 이미지는 레이저 스캐너(*Cyberware<sup>TM</sup>*)로 기록되어진 3차원 얼굴 모델의 데이터베이스(database)에서 얻어진 것이다. 이 이미지들은 256×256 픽셀 크기의 칼라 이미지이다. 그리고 얼굴의 머리카락을 이미지에서 완전하게 제거되

었다. 시뮬레이션에서는 칼라 이미지를 모두 8비트 흑백 이미지로 변환하였다. 시뮬레이션에 사용한 이미지는 모두 200개인데 임의로 99개의 데이터를 고른 후, 학습하는데 사용하였다. 그리고 테스트 입력은 나머지 101개의 데이터 중 하나를 골라서 인위적으로 눈, 코, 입 부위를 제거한 이미지를 사용하였다.

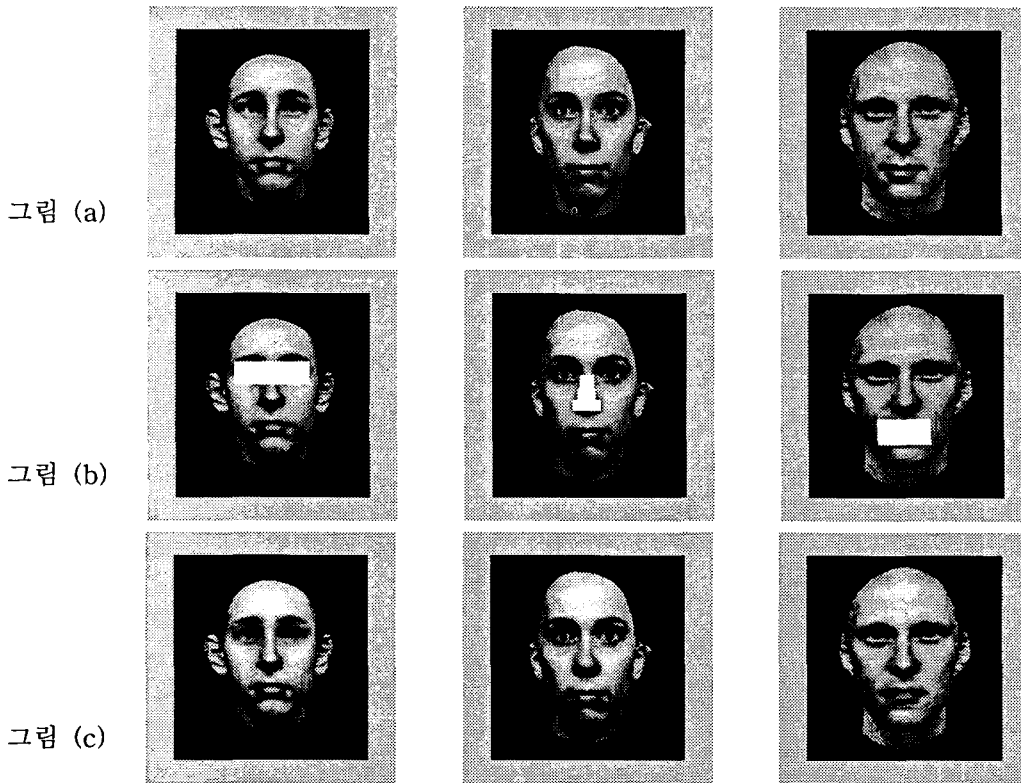
#### 3.2 부분 손상된 얼굴 영상의 결과

앞에서 언급한대로 모두 99개의 얼굴 영상을 학습 데이터 벡터로 사용하였다. 다음 페이지에 있는 얼굴 영상이 시뮬레이션 결과인데, 이 결과에 대해 설명을 하면 먼저 그림 (a)는 각 시뮬레이션에 사용한 얼굴 영상들의 원본을 보여준 것이고, 그림 (b) 각각 원본 얼굴 영상의 눈, 코, 입 부위를 인위적으로 제거한 이미지이다. 마지막으로 그림 (c)는 Kernel PCA 기법을 이용해서 노이즈를 제거한 뒤, 복원한 결과 이미지들이다. 하지만 복원한 결과 이미지가 대체적으로 조금 흐리게 나온다. 그래서 그 이미지를 사용하지 않고, 복원된 결과 이미지 중에서 가려진 부분에 대응하는 부분 이미지를 그림 (b)의 가려진 부분 이미지에 대체시켰다. 가려진 부분이 대체적으로 원본과 비슷하게 복원되었지만, 시뮬레이션 도중 그렇지 않은 얼굴도 복원되는 경우도 있었고, 흐리게 나온 부분도 있었다. 하지만 복원된 얼굴 영상이 대체적으로 자연스럽게 나왔다.

### 4. 결론 및 향후과제

본 논문에서는 부분적으로 손상된 얼굴 영상을 복원하는 기법에 대해서 소개했다. 만족스러운 결과가 나왔던 반면 문제점 역시 있었다. 현재 이 기법에서는 각각의 얼굴 이미지를 하나의 패턴 벡터로 학습하였는데, 이렇게 하면 앞에서 언급한대로 복원되는 부분의 이미지는 흐릿하게 나온다. 그 이유는 학습을 할 때, 픽셀 단위로 학습을 하였기 때문이다. 픽셀 단위로 학습했을 때 나쁜 특성이 나타나는 이유는 선형 공간의 중요한 특징 때문이다. 선형 공간의 가장 중요한 특징은 선형 결합을 정의하는 덧셈 법칙과 곱셈 법칙이 성립한다는 것이다. 간단한 예로 서로 다른 두 사람의 얼굴에 대해서 평균을 취하면, 나오는 얼굴은 두 얼굴을 겹쳐 놓은 것처럼 보인다. 비슷한 이유로, 본 논문에서 사용한 얼굴의 크기와 눈, 코, 입, 귀 등의 위치와 크기가 모두 다른 얼굴 영상에 대해 픽셀 단위로 학습하였기 때문에 결과 이미지가 흐리게 나온다. 이 문제점을 수정

1) 논문에서 사용한 얼굴 영상은 'Max Plank Institute for Biological Cybernetics'에서 제공받았다.



하기 위해서 향후에 얼굴 영상을 shape와 texture로 각각 나누어서 Kernel PCA 기법을 적용하려 한다. 새롭게 개발된 커널 기반 노이즈 제거 기법[10]의 적용도 주요한 향후 연구과제 중 하나이다. 그리고 실험하면서 사용한 얼굴이 모두 정면을 바로보고 있는데 위나 아래, 측면 등을 바라보는 얼굴에 대해서도 연구하려 한다.

### 참고문헌

- [1] N. Cristianini and J. Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods", Cambridge, U.K.: Cambridge University Press, 2000.
- [2] B. Schölkopf and A.J. Smola, "Learning with Kernels, Cambridge, MA: MIT Press, 2002
- [3] S. Mika, B. Schölkopf, A. Smola, K.R. Müller, M. Scholz, and G. Rätsch, "Kernel PCA and de-noising in feature space", Advances in Neural Information Processing Systems, vol. 11, pp. 536-542, Cambridge, MA: MIT Press, 1999
- [4] J.T. Kwok and I.W. Tsang, "The pre-image problem in kernel methods", IEEE Transactions on Neural Networks, vol. 15, pp. 1517-1525, 2004.
- [5] C.K.I. Williams, "On a connection between kernel PCA and metric multidimensional scaling", Machine Learning, vol. 46, pp. 11-19, 2002.
- [6] Vapnik, V. "Statistical learning theory", New York: Wiley, 1998
- [7] T.F.Cox and M.A.A.Cox, "Multidimensional Scaling", Monographs on Statistics and Applied Probability, vol. 88, 2nd Ed., London, U.K.: Chapman & Hall, 2001.
- [8] T.K.Moon and W.C.Stirling, "Mathematical Methods and Algorithms for Signal Processing", Upper Saddle River, NJ: Prentice Hall, 2000
- [9] B.W.Hwang and S.W.Lee."Reconstruction of partially damaged face images based on a morphable face model" IEEE Trans. on Pattern Analysis and Machine Intelligence, 25(3):365-372, March 2003
- [10] J.Park, D.Kang, J.Kim, J.T.Kwok, and I.W. Tsang, "Pattern de-noising based on support vector data description", To appear in the Proceedings of International Joint Conference on Neural Networks 2005.
- [11] <http://faces.kyb.tuebingen.mpg.de/>