

# Factorial Code 표현법을 이용한 얼굴 인식

정회원 이 오 영\*, 박 혜 영\*\*, 최 승 진\*\*\*

## Face Recognition via Factorial Code Representation

Oyoung LEE\*, Hyeyoung PARK\*\*, Seungjin CHOI\*\*\* *Regular Members*

### 요 약

얼굴인식에서 정보 이론적 접근방법은 얼굴 영상을 기저 영상의 합으로 분해하는 것을 기초로 한다. 가장 많이 쓰이고 있는 방법은 Principal Component Analysis (PCA)를 기반으로 하는 eigenface 방법이다. PCA를 기반으로 하는 방법은 데이터의 2차 통계적 구조만을 고려하므로 화소 사이의 고차 통계적 의존성은 고려되지 않는다. Factorial code 표현법은 효과적인 정보 표현의 좋은 방법으로 알려져 있고 이것은 Independent Component Analysis (ICA)와 밀접한 관련이 있다. Factorial code 표현법은 eigenface 방법과 비교할 때 중요한 정보가 포함되어 있는 데이터의 고차 통계적 구조도 고려되어 더욱 효과적인 정보 표현을 기대할 수 있다. 이 논문에서는 PCA를 이용하여 찾아낸 저차원 특징 공간에서 Factorial code 표현법을 이용하여 얼굴인식을 위한 통계적 특징점을 찾아낸다. 얼굴 인식에 있어서 Factorial code 표현법이 eigenface 방법보다 성능이 우수함을 모의실험을 통하여 입증한다.

### ABSTRACT

The information-theoretic approach to face recognition is based on the compact coding where face images are decomposed into a small set of basis images. A popular method for the compact coding may be the principal component analysis (PCA) which eigenface methods are based on. PCA based methods exploit only second-order statistical structure of the data, so higher-order statistical dependencies among pixels are not considered. Factorial coding is known as one primary principal for efficient information representation and is closely related to redundancy reduction and independent component analysis (ICA). The factorial code representation exploits high-order statistical structure of the data that contains important information and is expected to give more efficient information representation, compared to eigenface methods. In this paper we employ the factorial code representation in the reduced feature space found by the PCA and show that the factorial code representation outperforms the eigenface method in the task of face recognition. The high performance of the proposed method is confirmed by simulations.

### I. 서 론

얼굴인식은 사람마다 다른 얼굴 영상들의 데이터 베이스를 가지고 입력되는 얼굴 영상을 비교하여 누구인지 식별해 내는 것이다. 얼굴인식에서 정보 이론적 접근법은 얼굴 영상을 적은 수의 기저 영상의 합으로 분해하는 compact coding (under-

complete data representation)을 기반으로 하고 있다. 특징 벡터들은 기저 영상에 의해 생성된 subspace로 얼굴 영상을 사영시키므로 얻을 수 있다. 가장 널리 사용되는 방법은 factor analysis (FA) 또는 principal component analysis (PCA) 이다. Eigenpictures<sup>[15]</sup>, Eigenfaces<sup>[14,17]</sup>, Fisherfaces<sup>[5]</sup>, local feature analysis (LFA)<sup>[13]</sup> 등의 많은 얼굴인식

\* 현대중공업(주) 선박해양연구소 (loy250@korea.com)  
\*\*\* 포항공과대학교 컴퓨터공학과 (seungjin@postech.ac.kr)  
논문번호: 00370-0927, 접수일자: 2000년 9월 27일

\*\* 일본 이화학연구소 (hypark@brain.riken.go.jp)

알고리즘들이 PCA를 기반으로 하고 있다.

Compact coding은 최소 복원 오차를 갖는 축소된 수의 기저 벡터로 입력을 표현할 수 있는 선형 변환을 찾는 것이다. PCA는 잘 알려진 compact coding의 방법이다. PCA의 목적은 데이터의 최대 분산 방향을 나타내는 상호직교 기저 벡터의 집합을 찾는 것이고 여기서 기저 계수는 서로 상관관계가 없게 된다<sup>[11]</sup>. PCA는 mean square 관점에서 최적의 선형 coding으로 알려져 있다. 이러한 관점에서 차원 축소에 적합한 방법이지만 최적의 분류는 보장되지 않는다<sup>[9]</sup>. FA는 PCA와 밀접한 관계가 있고 목적은 선형 데이터 표현 즉, 데이터의 가장 좋은 공분산 구조 모델을 찾는 것이다. PCA나 FA 방법은 데이터의 2차 구조만을 고려하기 때문에 화소 사이의 고차 통계적 의존성은 고려되지 않았다.

Factorial coding은 효과적인 정보 표현에 적합한 하나의 원리로 알려져 있고 redundancy reduction과<sup>[3]</sup> independent component analysis (ICA)와도 밀접한 관계가 있다. Factorial code 표현은 데이터의 확률 분포를 가장 잘 나타내는 선형 데이터 표현을 찾는 것이 목적이다. 그러므로 고차의 통계적 구조도 고려된다. 이 논문에서는 latent variable model에서 factorial code 표현법을 얼굴인식에 적용한다. 영상 공간은 매우 큰 차원의 공간이므로 PCA 방법으로 찾아낸 축소된 저차원 특징 공간에서 factorial code를 이용 효율적인 데이터 표현을 위한 특징점을 추출한다. 모의실험을 통하여 제안된 방법의 우수성을 보이고 eigenface 방법과 비교한다.

## II. 특징 추출

### 2.1 Eigenface

패턴 인식에서 샘플 영상의 각각의 화소 값을 고차원 공간 (영상 공간)의 좌표로 다룬다.  $m_r \times m_c$ 의 얼굴 영상 집합,  $\{I_i\}_{i=1}^N$ 을 생각해 보자. 2차원의 영상  $I$ 는 벡터  $x$ 로 다루는 것이 편리하다. 영상 벡터  $x$ 는  $I$ 의 행을 계속 나열함으로 구성할 수 있고 학습영상은  $\{x_i\}_{i=1}^N$ 이 된다. 여기서 각각의  $x_i$ 는  $m$  ( $m_r \times m_c$ ) 차 벡터이다. 일반적으로 얼굴 영상은 매우 고차원 공간이므로 특징화 (characterization)나 분류화 (classification)를 위해 저차원 공간으로 표현하는 것이 요구된다. 얼굴 영상을 식별하기 위해 관계가 있는 적은 수의 기저 벡터만이 필요하다는 주장은 인간은 많은 수의 얼굴을 인식하고 저장할

수 있다는 것과, 이러한 인식이 빠른 시간 안에 순간적으로 이루어지므로 우리의 두뇌는 시각 정보를 처리하기 위해 이를 효과적인 형태로 부호화하는 방법을 사용할 것이라는 생각에 근거하고 있다. 수학적 관점과 인식의 관점 두 가지를 볼 때 PCA는 입력 데이터를 저차원의 공간으로 표현하는 효과적인 방법임에 틀림없다<sup>[15,17]</sup>.

PCA는  $m$  차의 영상 공간에서  $n$  차의 특징공간으로의 선형변환을 찾는다( $n < m$ ). 특징 벡터  $\{z_i\}$ 는 아래 식처럼 정의된다.

$$z_i = U(x_i - \mu), \quad (1)$$

여기서  $\mu$ 는 평균 영상 (그림 1. 참조)으로 아래 식과 같다.

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i. \quad (2)$$



그림 1. 학습 영상의 평균 영상.

그리고  $U \in \mathbb{R}^{n \times m}$ 는 정규 직교하는 행으로 구성된 선형 변환 행렬이다. PCA에서 행렬  $U$ 는 복원 오차가 최소가 되도록 선택된다. 행렬  $U$ 의 행들이  $\{x_i\}$ 의 샘플 공분산 행렬의 주 고유벡터이다<sup>[11]</sup>.

샘플 공분산 행렬  $C_x$  (scatter 행렬)은 아래 식처럼 주어진다.

$$\begin{aligned} C_x &= \frac{1}{N} \sum_{i=1}^N (x_i - \mu)(x_i - \mu)^T \\ &= \frac{1}{N} \Phi \Phi^T, \end{aligned} \quad (3)$$

여기서  $\Phi$ 는 행렬이고 각 열은 영상 벡터  $x_i$ 이다. 수식으로 표현하면 아래 식과 같다.

$$\Phi = [x_1 - \mu, \dots, x_N - \mu]. \quad (4)$$

샘플 공분산 행렬  $C_x$ 는 대칭성을 갖고있기 때문에 아래 수식처럼 eigen-decomposition 할 수 있다.

$$C_x = V \Lambda V^T, \quad (5)$$

여기서  $V$ 는 modal 행렬 ( $V$ 의 열 벡터는  $C_x$ 의 고유벡터에 대응된다)이고  $\Lambda$ 는 주 대각 요소가 큰 순서로 정렬된 고유값 행렬이다. PCA에서는 처음부터  $n$ 개의 열 벡터가 선택되고 이것을  $U$  행렬의  $n$ 개의 행 벡터와 같게 사용한다. 행렬  $U$ 의  $n$ 개의 행 벡터를 Eigenface (그림 4. 참조)라 부르고 기저 영상으로 사용된다.

### 2.2. Snap-Shot 방법

일반적으로 샘플 영상의 수  $N$ 은 영상 공간  $m$ 의 차원보다 작다. 이 경우에 샘플 공분산 행렬의 rank는 최대  $N$ 개이고  $m-N$ 개의 나머지 고유벡터는 null 공간에 속하게 된다. 첫 번째부터  $n$ 개의 고유벡터 ( $n \leq N < m$ )만이 중요하기 때문에 snapshot 방법으로 알려진 간단한 방법을 사용할 수 있다. 이 방법은  $C_x$ 의 고유 벡터를 직접 구하는 대신에 먼저  $\Phi^T \Phi$ 의 고유벡터를 찾는다. 행렬  $\Phi^T \Phi$ 의 고유벡터를  $\{e_i\}$ 라 가정하면

$$\Phi^T \Phi e_i = \lambda_i e_i. \quad (6)$$

양변에  $\Phi$ 를 앞쪽에 곱해주면 아래 식을 얻는다.

$$\Phi \Phi^T \Phi e_i = \lambda_i \Phi e_i. \quad (7)$$

위 식을 보면  $\Phi e_i$ 는  $\Phi \Phi^T$ 의 고유벡터임을 알 수 있다.

다른 방법으로는 고차의 공분산 행렬에서 처음 몇 개의 고유벡터를 계산하는 계산적으로 효율적인 방법인 EM 최적화를 사용하는 maximum likelihood PCA를 사용하여 고유벡터를 계산 할 수도 있다<sup>16)</sup>.

### 2.3. Fisherface

Eigenface 방법은 영상 공간에서 저차원의 특징 공간으로의 선형 사영을 기초로 하므로 모든 부류(class)의 전체를, 다시 말하면 전체 데이터 베이스의 모든 얼굴 영상을 최대화하는 사영 방향을 찾아낸다. 그렇기 때문에 조명 조건과 얼굴 표정의 변화로 생기는 원하지 않는 변화도 포함되게 된다. 이러한 이유로 Eigenface 방법은 저차원의 기저 벡터로부터 복원을 하는 관점에서는 최적의 방법이지만 식별(discrimination), 인식(recognition)의 관점에서서는 최적의 방법이 아니다<sup>5)</sup>.

패턴 인식에서 고전적 방법인 Fisher's linear

discriminant (FLD)를 고려해보자. FLD는 between-scatter 행렬과 within-scatter 행렬의 비율이 최대가 되는 선형 변환법이다<sup>9)</sup>. between-scatter 행렬  $S_B$ 는 아래 식과 같이 정의된다.

$$S_B = \sum_{k=1}^c N_k (\mu_k - \mu)(\mu_k - \mu)^T. \quad (8)$$

여기서  $N_k$ 는 class  $k$ 내의 데이터 수이고,  $\mu_k$ 는 class  $k$ 내에서 데이터의 샘플 평균이고,  $c$ 는 class의 수이다. within-scatter 행렬  $S_W$ 은 아래 식과 같다.

$$S_W = \sum_{k=1}^c \sum_{i \in C_k} (x_i - \mu_k)(x_i - \mu_k)^T. \quad (9)$$

FLD  $W_{FLD}$ 는 아래 식으로 얻을 수 있다.

$$W_{FLD} = \arg \max_W \text{tr} \{ (W S_W W^T)^{-1} (W S_B W^T) \}. \quad (10)$$

여기서  $\text{tr}$ 는 trace 연산자이다. 행렬  $W_{FLD}$ 의 열 벡터가 Fisherface가 된다.

### 2.4. Factorial Code 표현법

PCA에서 찾아낸 특징 벡터  $\{z_i\}$ 의 factorial code 표현법을 찾아보자. Linear generative model에서  $n$ 차의 특징 벡터  $z$ 는 아래 수식처럼 발생되었다고 가정할 수 있다.

$$\begin{aligned} z &= a_1 s_1 + \dots + a_n s_n, \\ &= A s, \end{aligned} \quad (11)$$

여기서  $A = [a_1, \dots, a_n] \in \mathbb{R}^{n \times n}$ 의 열 벡터는 기저 벡터 (기저 영상)이고 성분들  $s \in \mathbb{R}^n$ 은 기저 계수 (latent variables)라 한다.

Factorial code 표현법은 관측된 데이터의 확률 분포를 가장 잘 표현하는  $A$ 와  $s$ 를 찾는 것이 목적이다. 달리 말하면 관찰된 확률 밀도와 가장 잘 부합하는 모델 확률 밀도 즉 linear generative model (11)을 찾는 것이다<sup>7,8)</sup>. 관찰된 확률 밀도와 모델 확률 밀도를 각각  $p^o(z)$ 와  $p(z)$ 로 표시하고 기저 계수  $\{s_i\}$ 는 결합 확률 밀도 (joint density)가 주변 확률 밀도 (marginal densities)의 곱으로 표현할 수 있는 통계적 독립이라 가정하자.

$$p(s) = \prod_{i=1}^n p_i(s_i). \quad (12)$$

$p^o(z)$ 와  $p(z)$ 가 가장 잘 대응되는 행렬  $A$ 을 찾기 위하여 이들 사이의 Kullback-Leibler divergence를 생각해보면 risk  $R$ 은 아래 식처럼 표현된다.

$$\begin{aligned} R &= KL[p^o(z) \| p(z)] \\ &= \int p^o(z) \log \frac{p^o(z)}{p(z)} dz. \end{aligned} \quad (13)$$

latent variable 벡터  $s$ 로부터 특징 벡터  $z$ 로의 변환은 선형 (11) 이므로 변환 행렬의 Jacobian은 간단히 행렬식의 절댓값이 된다. 그러므로 아래 식과 같은 관계를 유도할 수 있다.

$$\log p(z) = -\log |\det A| + \sum_{i=1}^n \log p_i(s_i). \quad (14)$$

그러면 손실 함수  $L$ 은 다음 식처럼 된다.

$$L = \log |\det A| - \sum_{i=1}^n \log p_i(s_i), \quad (15)$$

여기서  $\log p^o(z)$ 는  $A$ 와 관계가 없으므로 생략된다.

Gradient descent 방법은  $A$ 에 관한 학습 알고리즘을 다음 식처럼 표현한다<sup>[3]</sup>.

$$\begin{aligned} \Delta A &= -\eta_i \frac{\partial L}{\partial A} \\ &= -\eta_i A^{-1} \{I - \varphi(s) s^T\}, \end{aligned} \quad (16)$$

여기서  $\eta_i > 0$ 는 학습률이고  $\varphi(s)$ 는 elementwise 함수로 다음과 같이 주어진다.

$$\varphi_i(s_i) = -\frac{d \log p_i(s_i)}{ds_i}. \quad (17)$$

매 학습마다 역행렬을 구하는 단점을 해결하기 위해 natural gradient 방법<sup>[1]</sup>을 사용한 학습 알고리즘은 다음과 같다.

$$\begin{aligned} \Delta A &= -\eta_i A A^T \frac{\partial L}{\partial A} \\ &= -\eta_i A \{I - \varphi(s) s^T\}. \end{aligned} \quad (18)$$

매 학습마다 기저 계수 벡터  $s$ 는 구해진  $A$ 를 이용하여  $s = A^{-1}z$ 를 계산하여 얻을 수 있다. 그

래서 얻어진  $s$ 로 다시  $A$ 를 구하는 과정을  $A$ 가 수렴할 때까지 반복한다.

비선형 함수  $\varphi_i(s_i)$ 는 prior  $\varphi_i(s_i)$ 와 관계가 있다. prior를 다르게 선택하면 다른 데이터 표현을 얻게 된다. 효과적인 정보 표현을 위하여 heavy-tailed 분포 (sparse 분포)를 갖는 super-Gaussian prior중의 하나인 Laplacian prior를 선택한다. Laplacian prior에서 비선형 함수  $\varphi_i(s_i)$ 는 아래 식의 형태를 갖는다.

$$\varphi_i(s_i) = \text{sgn}(s_i), \quad (19)$$

위 식에서  $\text{sgn}(\cdot)$ 은 signum 함수이다.

$A$ 를 학습시키는 대신에  $A^{-1}$ 를 학습시키는 방법도 가능하다.  $A^{-1}$ 는 ICA 필터와 부합된다<sup>[8,12]</sup>.  $W = A^{-1}$ 로 정의하면 natural gradient 방법을 이용한  $W$ 에 대한 학습 알고리즘은 다음과 같이 주어진다.

$$\Delta W = \eta_i \{I - \varphi(s) s^T\} W. \quad (20)$$

이것은 잘 알려진 ICA 알고리즘이다<sup>[2,4,6]</sup>.

### III. 인식 방법

#### 3.1. Nearest Neighbor

PCA를 이용한 축소시킨 특징 공간에서의 linear generative model은 학습되었고, factorial code 표현법에서의 새로운 특징 벡터  $s$ 는 다음과 같다.

$$s = A^{-1}U(x - \mu). \quad (21)$$

두 개의 특징 벡터들의 사이의 간단한 유사도 측정 방법은 Euclidean 거리를 계산하는 방법 (Nearest Neighbor)일 것이다. 테스트 영상에 대응하는 특징 벡터  $s_{test}$ 는 위의 식으로 계산 할 수 있다.  $s_{train}$ 와  $s_{test}$ 의 유사도는 다음 식으로 계산한다.

$$\epsilon_k = \|s_{train} - s_{test,k}\|^2. \quad (22)$$

위 식에서  $k$ 는 class의 수를 나타낸다.  $\epsilon_k$ 의 값이 최소값을 갖는 곳이 테스트 데이터가 속한 부류로 인식된다.

#### 3.2. 다층 신경망

인식 성능을 높이기 위해 다층 신경망을 이용한

인식방법에 대해 생각해보자.  $M$  개의 hidden node 와  $L$  개의 출력 node를 갖는 2층의 feedforward 신경망은 아래 식과 같은 형태의 deterministic 함수로 정의 될 수 있다<sup>[7]</sup>.

$$f_i(z; \theta) = \sigma_o \left\{ \sum_j v_{ij} \sigma_h(w_j \cdot z + b_j) + c_i \right\}, \quad (23)$$

여기서 각각의  $f_i(z; \theta)$  는  $i$  번째 출력 node와 대응되고  $i = 1, \dots, L$ 이다. 매개변수 벡터  $\theta$  는 모든 매개변수  $w_j, v_{ij}, b_j, c_i$  를 모두 포함한다. Hidden node의 activation 함수  $\sigma_h(\cdot)$  는 sigmoid 함수이고 모든 출력 node의 값을 더한 것이 unity 가 되게 하기 위해 출력 node에서의 activation 함수  $\sigma_o(\cdot)$  는 다음과 같이 정의한다.

$$\sigma_o(net_i) = \frac{\exp(net_i)}{\sum_k \exp(net_k)}, \quad (24)$$

$$net_i = \sum_j v_{ij} \sigma_h(w_j \cdot z + b_j) + c_i. \quad (25)$$

입력 특징 벡터  $z$  가 주어졌을 때 랜덤 벡터  $y$  의 conditional probability를 추정하기 위해 이 신경망 모델을 이용한다. 대응하는 conditional 확률 밀도 함수는 아래 식과 같이 표현할 수 있다.

$$p(y|z; \theta) = \prod_i f_i(z; \theta)^{y_i}. \quad (26)$$

여기서 랜덤 벡터  $y$  는 입력  $z$  가 어느 class에 속하는지 아래의 방법으로 표시한다.

$$y_i = \begin{cases} 1 & i = \arg \min_i f_i(z; \theta) \\ 0 & \text{otherwise} \end{cases} \quad (27)$$

최적의 값을 갖는 매개변수  $\theta$  를 찾기 위해 stochastic gradient descent 학습법을 사용하자. 이것은 log-likelihood 함수를 최대화하는 것이다.

$$l(z, y^*; \theta) = \sum_{j=1}^L y_j^* \log f_j(z; \theta). \quad (28)$$

최종 학습 알고리즘은 아래와 같이 표현된다.

$$\theta_{i+1} = \theta_i + \eta_i \frac{\partial l(z_i, y_i^*; \theta_i)}{\partial \theta_i}. \quad (29)$$

위 식에서  $\eta_i$  는 학습률이다.

#### IV. 실험결과

PCA를 기반으로 하는 방법과 factorial code 표현법의 얼굴 인식 성능을 평가하기 위해 AR face database<sup>[10]</sup>를 이용하였다. Database의 데이터는 40 명의 각각 다른 얼굴 표정 (자연스런 표정, 웃는 표정, 화난 표정)과 다른 조명 조건 (왼쪽 조명, 오른쪽 조명)의 정면 얼굴 영상을 2 session(2주의 차이를 두고 얻은 영상)으로 구성되어있다. 모두 400개의 얼굴 영상(40명×5장×2 session)중에 200개를 선택 (첫 번째 session에서 자연스런 표정, 화난 표정, 오른쪽 조명, 두 번째 session에서 웃는 표정, 왼쪽 조명)하여 학습에 사용하였다. 나머지 200장의 얼굴 영상 (첫 번째 session에서 웃는 표정, 왼쪽 조명, 두 번째 session에서 자연스런 표정, 화난 표정, 오른쪽 조명)은 테스트 영상으로 사용하였다. 각 얼굴 영상은 46×50 화소의 크기를 갖는다. 얼굴 영상은 행을 계속 나열한 2300×1의 벡터로 만들어 사용하였다. 그림 2.과 그림 3.은 학습 영상과 테스트 영상의 표본이다.



그림 2. 학습 영상으로 사용한 얼굴 영상 표본.



그림 3. 테스트 영상으로 사용한 얼굴 영상 표본.

PCA 방법을 학습 영상에 적용하여 선형 변환 행렬  $U$  를 찾아내었다. 특징 벡터  $z \in R^n$  은 다음 식으로부터 구하였다.

$$z = U(x - \mu) \quad (30)$$

선형 변환 행렬  $U$  의 행 벡터에 대응되는 첫 번째 부터 20개의 eigenfaces를 그림 4.에 나타내었다. 얼굴 인식은 축소된 특징 공간 ( $n = 20, n = 30$ ) 에 nearest neighbor 방법을 사용하여 수행하였다. 그리고 학습을 통하여 인식을 수행하는 신경회로망 방법을 축소된 특징 공간 ( $n = 20, n = 50$ )에 적용하여 nearest neighbor 방법과 신경회로망 방법

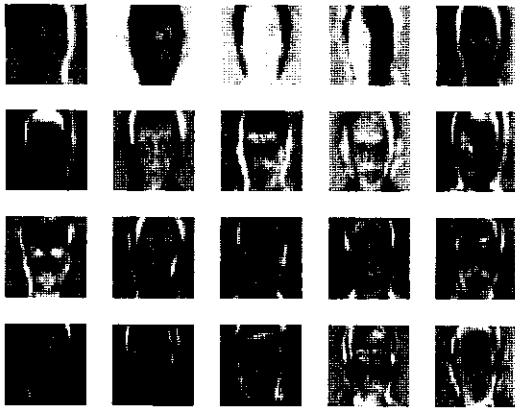


그림 4. Eigenface 방법을 적용하여 찾아낸 20개의 기저 영상 (왼쪽에서 오른쪽으로 위에서 아래로 정렬).



그림 5. Factorial code 표현법을 적용하여 찾아낸 20개의 기저 영상 (왼쪽에서 오른쪽으로 위에서 아래로 정렬).

의 인식 성능을 비교하였다.

Factorial code 표현법을 Eigenface 방법으로 찾은 특징 벡터  $z$  에 적용하였고 얻어진 20개의 기저 영상을 그림 5.에 나타내었다. Principal component 특징 벡터  $z$  는 factorial code 표현법을 사용하여 찾아낸 기저 영상위로 사영시켰다. 새로운 특징 벡터에 nearest neighbor 방법을 사용하여 얼굴 인식을 수행하였다. 실험 결과 (그림 6. 7.)에서 잘 나타나듯이 factorial code 표현법을 사용한 것이 Eigenface 방법이나 Snap-shot 방법만을 사용한 것보다 성능이 우수함을 확인할 수 있다.

다른 실험으로 FLD 방법을 축소된 특징 공간에 적용하였다. 이 방법 또한 성능의 개선을 가져왔다. 20개, 30개의 특징 벡터를 가지고 수행한 결과 Eigenface 방법이나 Snap-shot 방법 사용 후 FLD를 적용한 방법이 좋은 성능을 보임을 알 수 있다. 그리고 FLD를 적용한 후 factorial code 표현법을 적용한 방법의 성능이 가장 우수함을 볼 수 있다 (그림 8. 9. 참조). 이것은 Eigenface 방법이나 snap-shot 방법과는 다르게 조명 조건과 얼굴 표정의 변화로 생기는 원하지 않는 변화를 FLD 방법을 이용하여 어느 정도 해결할 수 있음을 보여준다. 그리고, FLD 방법을 factorial code 표현법을 적용한 후에도 적용하여 보았으나 인식 성능에 향상을 가져오지는 못했다. 이것은 factorial code 표현법이 데이터의 고차의 통계적 구조도 고려하는 반면에 FLD는 2차의 구조만을 고려하기 때문이다.

단순히 Euclidean 거리를 계산하여 인식하는 것보다는 신경회로망을 이용하여 인식하는 것이 인식 성능면에서 우수함을 알 수 있다 (그림 10. 11. 참조).

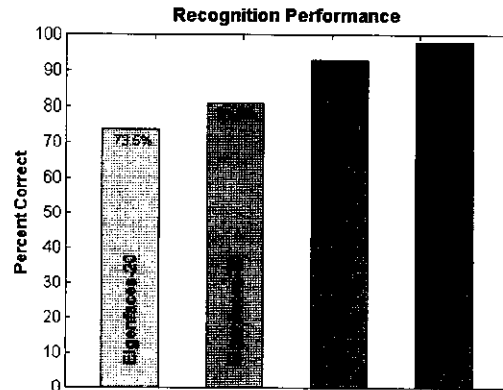


그림 6. Eigenface 방법으로 찾아낸 20개와 30개의 특징벡터에 nearest neighbor classifier를 이용하여 factorial code 표현법과의 인식 성능 비교

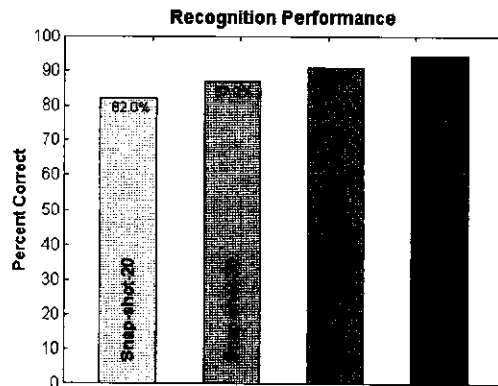


그림 7. Snap-Shot 방법으로 찾아낸 20개와 30개의 특징 벡터에 nearest neighbor classifier를 이용하여 factorial code 표현법과의 인식 성능 비교

그림 6. 7. 8. 9. 10. 11.에 나타난 전체적인 실험 결과에서 볼 수 있듯이 데이터의 고차 통계적 구조까지 고려하는 factorial code 표현법을 적용한 방법이

데이터의 2차 통계적 구조만을 고려하는 Eigenface 방법, Snap-shot 방법, 그리고 Fisherface 방법과 비교해 볼 때 좋은 성능을 나타냄을 알 수 있다.

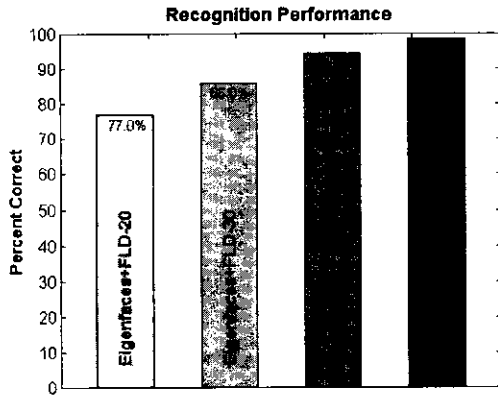


그림 8. Eigenface 방법으로 찾아낸 20개와 30개의 특징벡터에 FLD를 적용하고 nearest neighbor classifier를 이용하여 factorial code 표현법과의 인식 성능 비교

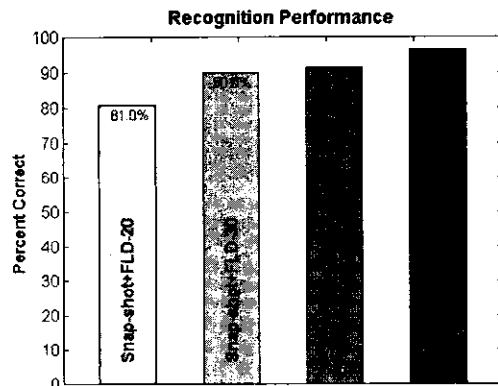


그림 9. Snap-shot 방법으로 찾아낸 20개와 30개의 특징 벡터에 FLD를 적용하고 nearest neighbor classifier를 이용하여 factorial code 표현법과의 인식 성능 비교

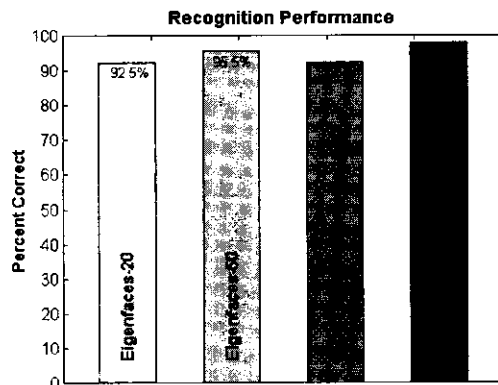


그림 10. Eigenface 방법으로 찾아낸 20개와 50개의 특징 벡터에 MLP classifier를 이용하여 factorial code 표현법과의 인식 성능 비교

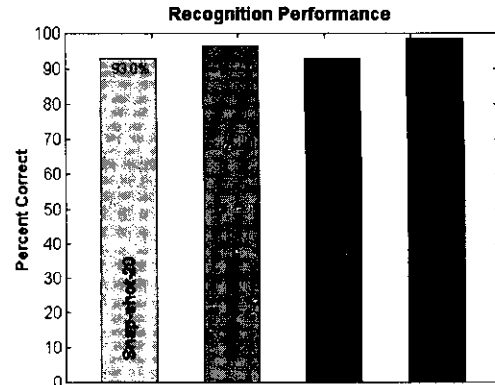


그림 11. Snap-shot 방법으로 찾아낸 20개와 50개의 특징 벡터에 MLP classifier를 이용하여 factorial code 표현법과의 인식 성능 비교

### V. 결론 및 토의

얼굴 인식에서 정보 이론적인 접근 방법은 얼굴 영상을 기저 영상으로 분해하고 얼굴 영상을 기저 영상에 사영하여 특징 벡터를 얻어낸다. 이 논문에서 factorial code 표현 방법을 적용하여 기저 영상을 얻었다. Factorial code 표현 방법의 성능이 eigenface와 fisherface 방법만을 적용한 것보다 우수함을 보였다. Eigenface와 snap-shot 방법을 이용하여 찾아낸 200개의 특징 벡터들 중에서 20개, 30개, 50개의 적은 수의 특징 벡터만을 사용하여도 성능이 보장됨을 보였다. 이러한 결과는 2차의 통계적 구조 즉, 평균 (mean)과 분산(variance)만을 고려하는 eigenface 방법, snap-shot방법, fisherface 방법과는 다르게 더 많은 중요한 정보를 포함하는 데이터의 고차 통계적 구조까지 고려하는 factorial code 표현의 특성에 기인한다. 그리고, factorial code 표현 방법이 ICA 방법을 기초로 하는 비선형 정보 최대화의 독립 요소 표현 방법과 유사함을 보였다.

인식 방법에서는 단순히 특징 벡터들 사이의 Euclidean 거리만을 계산하는 nearest neighbor 보다는 학습을 통해 유사함을 찾아내는 신경회로망 방법이 우수함을 보였다.

이 논문에서 실험을 위해 사용되었던 영상 데이터들이 모두 정면 영상에 한정 되어있었다. Factorial code 표현 방법이 데이터의 고차 통계적 구조까지 포함한다 하더라도 factorial code 표현 방법의 유용성이 전체적 선형성에 제한되어 있었다. 얼굴의 각도가 변하는 영상에서의 얼굴 인식을 하기 위하여 view-based eigenspaces와 factorial coding을 병합하는 것이 필요하고, 또한 local 선

형 모델을 사용하는 것도 필요하다.

### Acknowledgment

본 연구는 과기부의 뇌신경정보학 사업단 과제와 국제공동연구, 그리고 두뇌한국 21 사업에 의하여 수행되었음.

### 참 고 문 헌

[1] S. Amari. "Natural gradient works efficiently in learning". *Neural Computation*, 10(2):251-276, Feb. 1998.

[2] S. Amari, T. P. Chen, and A. Cichocki. "Stability analysis of learning algorithms for blind source separation". *Neural Networks*, 10(8):1345-1351, 1997.

[3] H. B. Barlow. "Unsupervised learning". *Neural Computation*, 1:295-311, 1989.

[4] M. S. Bartlett, H. M. Lades, and T. J. Sejnowski. "Independent component representations for face recognition". In *Proceedings of the SPIE Symposium on Electronic Imaging: Science and Technology; Conference on Human Vision and Electronic Imaging III*, pages 528-539, San Jose, California, Jan. 1998.

[5] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection". *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):711-720, Oct. 1997.

[6] A. Bell and T. Sejnowski. "An information maximization approach to blind separation and blind deconvolution". *Neural Computation*, 7:1129-1159, 1995.

[7] S. Choi and O. Lee, "Factorial code representation of faces for recognition," *Lecture Notes in Computer Science 1811, Biologically Motivated Computer Vision*, S. -W. Lee, H. H. Bulthoff, and T. Poggio, Eds., pp. 42-51, Springer, 2000.

[8] S. Choi and O. Lee, "Sparse independent component analysis: Efficient coding of natural

images?", *Journal of Electrical Engineering and Information Science*, vol. 5, no. 2, pp. 84-92, Apr. 2000.

[9] K. Fukunaga. "An Introduction to Statistical Pattern Recognition". Academic Press, New York, NY, 1990.

[10] A. Martinez and R. Benavente. "The AR face database. Technical Report CVC #24", Computer Vision Center, Purdue University, June 1998.

[11] E. Oja. "Neural networks, principal component analysis, and subspaces". *International Journal of Neural Systems*, 1:61-68, 1989.

[12] B. A. Olshausen and D. J. Field. "Emergence of simple-cell receptive field properties by learning a sparse code for natural images". *Nature*, 381:607-609, 1996.

[13] P. Penev and J. Atic. "Local feature analysis: A general statistical theory for object representation". *Network: Computation in Neural Systems*, 7(3):477-500, 1996.

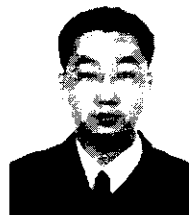
[14] A. Pentland, B. Moghaddam, and T. Startner. "View-based and modular eigenspaces for recognition". In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1994.

[15] L. Sirovich and M. Kirby. "Low-dimensional procedure for the characterization of human faces". *Journal of the Optical Society of America A*, 4(3):519-524, 1987.

[16] M. E. Tipping and C. M. Bishop. "Mixture of probabilistic principal component analyzers". *Neural Computation*, 11(2):443-482, 1999.

[17] M. Turk and A. Pentland. "Eigenfaces for recognition". *Journal of Cognitive Neuroscience*, 3(1):71-86, 1991.

이 오 영(Oyoung LEE)



1998년 2월 : 충북대학교  
전기공학과 공학사  
2001년 2월 : 충북대학교  
전기공학과 대학원 석사  
2001년 1월~현재 : 현대중공업  
(주) 선박해양연구소  
연구원



<주관심 분야> Independent component analysis,  
얼굴인식, 문자인식.

박 혜 영(Hyeyoung PARK)



1994년 2월 : 연세대학교  
컴퓨터과학과 학사  
1996년 2월 : 연세대학교  
컴퓨터과학과 석사  
2000년 2월 : 연세대학교  
컴퓨터과학과 박사

2000년 3월~현재 : 일본 RIKEN 연구소 연구원.

<주관심 분야> 신경회로망, 기계학습, 정보이론, 영  
상처리, 생체정보학.

최 승 진(Seungjin CHOI) 정회원



1987년 2월 : 서울대학교  
공과대학 전기공학과  
공학사  
1989년 2월 : 서울대학교 대학원  
전기공학 석사  
1996년 8월 : 미국 Notre Dame  
Univ. 대학원 공학박사,

1997년 8월 : 일본 RIEKN 연구소 선임연구원

1997년 8월~2001년 2월 : 충북대학교 전기전자공학  
부 조교수

1997년 12월~현재 : 일본 RIEKN 연구소 초빙 연구원

2001년 2월~현재 : 포항공과대학교 컴퓨터공학과 조  
교수.

<주관심 분야> Probabilistic/Statistical Learning,  
Neural Computation, Pattern Recogni-  
tion, Brain Computer Interface, Bioin-  
formatics.