

# 실시간 얼굴 검출 시스템 설계 및 구현

이호근<sup>†</sup>, 정성태<sup>\*\*</sup>

## 요 약

본 논문에서는 웹카메라 영상과 같은 저해상도의 동영상으로부터 실시간으로 다중 얼굴을 검출할 수 있는 시스템을 제안한다. 본 논문에서는 먼저 영상내의 거대한 특징 집합으로부터 중요한 작은 특징 집합을 선택하는 AdaBoost 기반 객체 검출 방법을 사용하여 얼굴 후보 영역을 검출한다. 검출된 얼굴 후보 영역에 대한 주성분 분석을 수행함으로써 데이터의 크기가 현저히 줄어든 특징 벡터를 구한다. 그 다음에는 특징 벡터에 대해 SVM 기반 이진분류를 수행하여 후보 영역의 영상이 얼굴인지 아닌지를 판별한다. 실험결과에 의하면, 본 논문에서 제안한 방법은 저해상도 동영상에서 실시간 처리가 가능한 다중 얼굴 검출 성능을 보였고, 주성분 분석과 SVM을 이용한 얼굴 검증 과정을 통해 얼굴 검출의 정확도를 향상 시켰다.

## Design and Implementation of a Real-Time Face Detection System

Ho-Geun Lee<sup>†</sup>, Sung-Tae Jung<sup>\*\*</sup>

## ABSTRACT

This paper proposes a real-time face detection system which detects multiple faces from low resolution video such as web-camera video. First, It finds face region candidates by using AdaBoost based object detection method which selects a small number of critical features from a larger set. Next, it generates reduced feature vector for each face region candidate by using principle component analysis. Finally, it classifies if the candidate is a face or non-face by using SVM(Support Vector Machine) based binary classification. According to experiment results, the proposed method achieves real-time face detection from low resolution video. Also, it reduces the false detection rate than existing methods by using PCA and SVM based face classification step.

**Key words:** Face Detection(얼굴 검출), AdaBoost(아다부스트), PCA(주성분 분석), SVM(서포트 벡터 머신)

## 1. 서 론

최근 들어 컴퓨터 비전은 다양한 분야에서 활발하게 연구가 진행되어 왔다. 특히, 얼굴 검출은 영상

감시 시스템, 원격회의 시스템, 휴먼-컴퓨터 인터페이스, 얼굴 인식 시스템과 같은 응용을 위한 핵심 요소 기술로서 널리 연구되고 있다. 얼굴 검출은 얼굴의 움직임, 크기 변화, 얼굴 표정, 인종, 나이, 성별, 헤어스타일, 타 객체와의 겹침, 조명, 카메라의 기계적인 특성 등에 따라 다양하게 나타날 수 있기 때문에, 얼굴 검출 기술에 대한 상당한 연구가 진행되었지만 아직도 많은 문제점을 가지고 있다. 기존의 얼굴 검출 방법은 지식 기반(knowledge-base) 방법[1], 특징 기반(feature-based) 방법[2], 형판 정합(template matching) 방법[3], 외형 기반(appearance-base) 방법[4] 등이 있다.

※ 교신저자(Corresponding Author) : 이호근, 주소 : 전북북도 익산시 신용동 344-2(570-749), 전화 : 063)850-6886, FAX : 063)856-8009, E-mail : hghome@wonkwang.ac.kr  
접수일 : 2004년 11월 18일, 완료일 : 2005년 2월 25일

<sup>†</sup> 준회원, 아이디어스 IS평가기술구름 근무  
<sup>\*\*</sup> 종신회원, 원광대학교 전기전자및정보공학부 교수  
(E-mail : stjung@wonkwang.ac.kr)

※ 본 논문은 2004년도 과학기술부 지원 지방연구 중심대학 육성사업에 의해 수행되었음.

기존의 얼굴 검출 시스템에서 얼굴 스킨 색을 이용한 방법과 모션 정보를 이용하여 배경을 제거하는 방법은 조명과 카메라 등 외부환경에 따라 매우 제한적 시스템이고, 신경망 등은 많은 연산 작업을 필요로 한다. 대부분의 스틸 이미지에 대한 얼굴 검출 시스템은 높은 해상도의 이미지를 이용하여 처리하며, 결과에 대한 빠른 응답을 요구하지 않고 정확한 검출 성능에 초점이 맞추어져 있다. 그러나 비디오 영상 또는 카메라를 이용한 시스템은 낮은 해상도에서 빠른 시간 내에 변하는 이미지의 정보를 이용하여 얼굴 검출 정확성과 함께 빠른 응답을 요구하게 된다. 따라서 본 논문에서는 낮은 해상도의 저가형 웹카메라에서 실시간 다중 얼굴 검출을 위해 이미지내의 픽셀들의 정보에서 특징(feature) 정보를 생성하고 이를 이용하는 방법을 사용하였다. 이러한 특징 정보를 이용하여 얼굴 영역을 검출하는 과정의 가장 큰 문제점은 주어진 이미지내에 존재하는 다양한 크기의 얼굴을 찾기 위해 멀티스케일의 검색 윈도우를 이용하여 주어진 이미지 상에서 쉬프트 하면서 스캔 과정을 수행하는데 이는 많은 연산을 필요로 한다. 이를 해결하기 위해 빠르게 객체를 검출하기 위한 AdaBoost 객체 검출 방법이 제안되었다[5-7]. AdaBoost 방법은 영상내의 거대한 특징 집합으로부터 중요한 작은 특징 집합을 선택하여 연산량을 줄이는 방법으로 Haar-like 특징과 인테그랄 이미지를 적용하여 빠르게 정확한 얼굴 후보 영역을 검출한다. 본 논문에서는 AdaBoost에 의해 검출된 얼굴 영역의 정확성을 향상시키기 위해 얼굴 검증과정을 추가하였다. 검출된 얼굴 영역에 대해 주성분 분석[14]을 수행함으로써 데이터의 크기가 현저히 줄어든 특징 벡터를 추출하고, 추출된 특징 벡터에 대해 SVM 기반 이진분류[8,9]를 수행하여 얼굴 영역의 영상이 얼굴인지 아닌지를 판별하였다. 또한 팬(Pan)-틸트(Tilt) 제어가 가능한 저가형 웹카메라를 이용하여 자동으로 얼굴 위치를 추적, 이동하면서 얼굴 검출을 수행하였다. 본 논문에서 제안하는 다중 얼굴 검출 시스템의 실험 결과에 의하면, 얼굴 검출 시간이 평균 21ms로 기존의 시스템보다 상당히 빠른 속도로 연산을 수행하였고, 이는 동영상으로부터 다중 얼굴 검출의 실시간 처리가 가능함을 보였다. 또한 AdaBoost 방법에 의해 얼굴 후보를 검출한 경우에는 오인식된 영역을 검증하기 위해 PCA를 이용하여 얼굴의 주성분을 분

석하고 얼굴에 대한 특징 계수 60개로 감소시키고 SVM 얼굴 검증을 수행할 때, 검출 시간은 약 0.1ms로 거의 변화가 없었으나, 오검출률이 3.80%에서 0.22%로 얼굴 검출의 정확도가 향상되었다. 이는 SVM 단독으로 사용할 때보다 PCA+SVM 방법이 검출 시간과 검출률이 훌륭하였다. 또한 팬-틸트 웹카메라를 이용하여 움직이는 얼굴을 추적하므로 얼굴 검출의 성능이 향상되었다.

## 2. 간단한 특징을 이용한 빠른 얼굴 후보 영역 검출

### 2.1 전처리 단계

본 논문에서는 얼굴 후보 영역을 검출하기 위해 입력 영상에 대한 전처리 단계로 조명 보정, 히스토그램 평활화, 모션 추적 과정을 적용하였다. 조명 보정은 극단적인 빛(light)의 각도에 따른 격심한 그림자를 줄이기 위해 사용하였다. 히스토그램 평활화는 이미지내의 명암 값의 분포가 특정 값이 집중되거나 균일하지 못한 영상에 대해 명암 값이 일정한 분포를 갖도록 하여 이미지내의 조명 밝기의 차이와 카메라의 노출에 따른 응답곡선을 보정하기 위해 사용하였다. 모션 추적은 동영상 내에서 움직임이 없는 경우에는 얼굴 검출을 수행하지 않기 위해서 움직임이 발생한 경우에만 얼굴 검출 수행하였다. 실시간 얼굴 검출에서 불필요한 얼굴 검출 수행에 따른 재계산을 수행하지 않기 위해서 적용하였다. 모션 추적을 검출하기 위해서는 이전 프레임과 현재 프레임의 차이를 이용하였고, 움직임이 발생한 경우에 대하여 얼굴 후보 영역 검출을 수행하였다.

### 2.2 Haar-like 특징과 인테그랄 이미지

카메라로부터 입력되는 얼굴 영상은 다양한 외부 환경 요소에 영향을 많이 받는다. 이와 같은 환경 요소에 강인하면서 빠르게 연산이 가능한 얼굴 검출 방법이 요구된다. 이를 위해 그림 1과 같이 얼굴 영역 내에서 간단하고 객체 간의 구별 능력이 있는 특징을 사용하여 해결한다.

본 논문에서 얼굴 검출을 위한 특징으로 Papa-georiou et al[5,6]에 의해 제안된 간단하면서도 연산이 빠른 Haar-like 특징을 얼굴 검출을 위한 특징

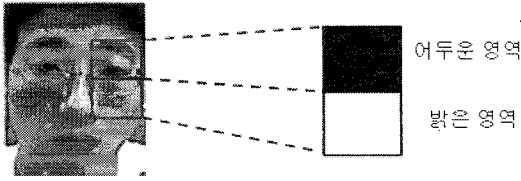


그림 1. 얼굴 영역 내의 특징 예

집합으로 사용하였고, Haar-like 특징은 인테그랄 이미지를 이용하여 빠르게 연산 할 수 있다. 이러한 Haar-like 특징과 인테그랄 이미지의 사용은 복잡한 연산을 줄이고, 높은 검출 성능이 요구는 실시간 얼굴 검출에 효과적으로 적용된다.

그림 2에서의 Haar-like 특징 프로토타입들은 이미지내의 얼굴을 검출하기 위한 효과적인 특징으로 사용된다. Haar-like 특징 계수를 이용해서 얼굴 영역을 검출하면, 유동적인 얼굴의 특성에도 강인한 검출 성능을 보인다. 이미지 내에서 Haar-like 사각형의 픽셀 합계 계산을 빠르게 수행하기 위해 인테그랄 이미지를 사용하였다. 인테그랄 이미지는 영상처리에서 사용하는 영역 합 테이블(Summed Area Table: SAT)과 유사하다. SAT는 중복연산을 최소화하여 연산 속도를 빠르게 한다.

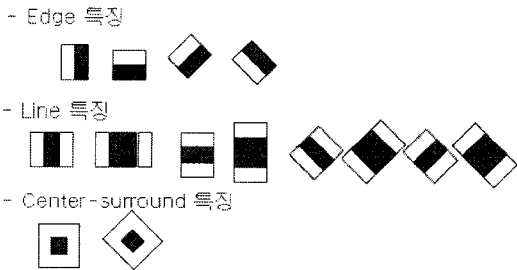


그림 2. Haar-like 특징 프로토타입

그림 3(a)의 SAT는 입력영상의 원점(0,0)에서부터 좌표 영역  $(x,y)$ 를 포함하는 사각형 영역의 모든 픽셀 값의 합이며, (1), (2)와 같이 계산된다. 그림 3(b)에서 좌측 상단의 좌표가  $(x,y)$ 이고 너비와 높이가 각각  $w$ 와  $h$ 이고 회전각도가  $\alpha$ 인 사각형을  $r=(x,y,w,h,\alpha)$ 로 나타내고, 사각형 영역  $r$ 의 픽셀 값의 합을  $RecSum(r)$ 로 표시한다.  $RecSum(r)$ 의 값은 SAT를 이용하여 효과적으로 계산될 수 있다. 회전각도  $\alpha=0$ 인 사각형 영역  $r=(x,y,w,h,0)$ 에 대한  $RecSum(r)$ 의 값은 그림 3(b)에 나타나 있는 바와 같이 (3)에 의해 계산된다.

$$SAT(x,y) = \sum_{x' < x, y' < y} I(x',y') \quad (1)$$

$$SAT(x,y) = SAT(x,y-1) + SAT(x-1,y) + I(x,y) - SAT(x-1,y-1) \quad (2)$$

$$RecSum(r) = SAT(x-1,y-1) + SAT(x+w-1,y+h-1) - SAT(x-1,y+h-1) - SAT(x+w-1,y-1) \quad (3)$$

이미지 내에서 45° 회전한 사각형 RSAT는 사각형 영역의 모든 픽셀 값을 합이며, (4), (5)와 같이 계산되고, RSAT를 이용하여 특정 위치에 존재하는 사각형 영역  $r=(x,y,w,h,45)$ 의 합계  $RecSum(r)$ 은 그림 4(b)에서와 같이 (6)에 의해 계산된다.

$$RSAT(x,y) = \sum_{y' < y, y' \leq y - |x-x'|} I(x',y') \quad (4)$$

$$RSAT(x,y) = RSAT(x-1,y-1) + RSAT(x+1,y-1) - RSAT(x,y-2) + I(x,y) + I(x,y-1) \quad (5)$$

$$RecSum(r) = RSAT(x-h+w,y+w+h-1) + RSAT(x,y-1) - RSAT(x-h,y+h-1) - RSAT(x+w,y+w-1) \quad (6)$$

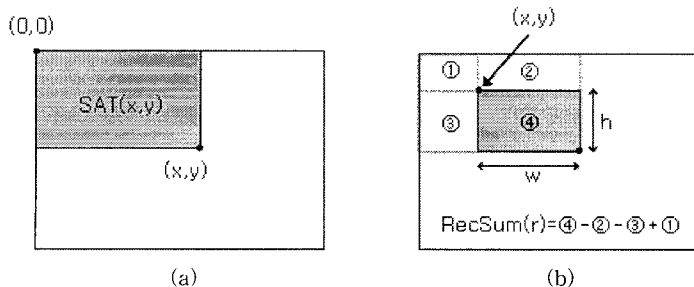


그림 3. 사각형의 인테그랄 이미지: (a) SAT, (b) 사각형내의 픽셀 합계.

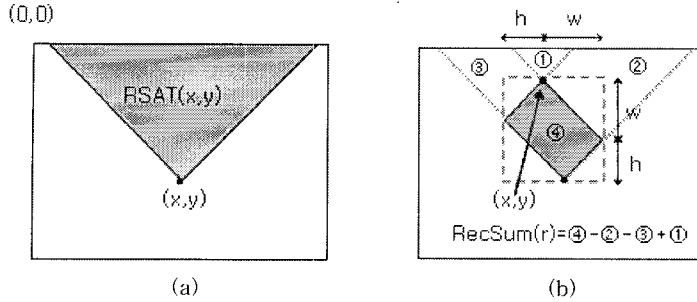


그림 4. 45° 회전 사각형의 인테그랄 이미지: (a) RSAT, (b) 사각형내의 픽셀 합계.

### 2.3 AdaBoost 알고리즘을 이용한 얼굴 분류기 생성

Haar-like 특징 추출 단계에서 생성된 얼굴 특징 계수를 이용하여 얼굴을 분류하기 위해서 본 논문에서는 Adaptive Boosting(AdaBoost) 기법과 캐스케이드(cascade) 구조를 이용하였다. AdaBoost 알고리즘은  $t$ 개의 약한분류기의 선형적인 결합을 통하여 최종적으로 높은 검출 성능을 가지는 강한분류기를 생성하여 분류하는 기법이다. Freund와 Schapire에 의해 소개 되었으며 에러율은 약한분류기의 수가 증가할수록 영점에 지수적으로 근접함이 증명되었다 [5-7]. AdaBoost 알고리즘은 기본적으로 데이터의 확률 분포를 가지고 학습이 진행된다. 현재 약한분류기를 학습 시킨 후 올바르게 분류된 데이터에 대해서는 확률 분포를 낮추고 올바르게 안게 분류된 데이터에 대해서는 확률 분포를 높인다. 그러면 다음 번 약한분류기를 만들어 낼 때는 잘못 분류된 데이터에 집중해서 학습이 된 약한분류기가 탄생하게 된다.

AdaBoost 알고리즘을 이용한 얼굴 검출 방법은 얼굴과 비얼굴 패턴으로 구성된 훈련영상 집합에서 추출된 사각형 특징 집합을 이용하여 학습된다. 오분류된 훈련 패턴은 가중치를 증가시키고, 바르게 분류된 훈련 패턴은 가중치를 감소시키는 과정을 반복하면서 최소의 에러율을 나타내는 Haar-like 특징만을 선택한다. 약한분류기는 1개의 사각형 특징으로 분류함을 의미하며 강한분류기는  $t$ 개의 사각형 특징을 선형적 결합을 통해 분류함을 의미한다. 즉, 분류기의 가중치를 수정해 가면서 단계적으로 반복하여 각 단계마다 특징의 선형 집합 중에서 얼굴을 검출하는데 결정적인 역할을 하는 특징만을 남기고 나머지는 제거하는 방식이다.

### 2.4 캐스케이드(cascade) 구조를 이용한 빠른 배경 제거

본 논문에서는 얼굴 검출의 연산 속도를 향상시키기 위해 그림 5와 같이 AdaBoost 분류를 캐스케이드 구조로 만들어 적용하였다. 캐스케이드 구조는 여러 개의 스테이지를 나누고 스테이지별 약한분류기의 수를 다르게 하여 수행하는 방법이다. 처음 스테이지 분류에서 대부분의 배경(비얼굴)을 제거되는 기능을 가지고 있어 단일 스테이지를 적용하는 방법보다 훌륭한 성능을 보였다.

본 논문에서 AdaBoost 학습을 위해 총 24개의 스테이지를 이용하였고, AdaBoost 알고리즘을 이용한 얼굴 검출 학습의 결과로 그림 6에 첫 번째 특징(약한분류기)과 두 번째 특징(약한분류기)의 예가 표시되어 있다. 첫 번째 특징은 눈 영역과 뺨 영역의 다른 명도가 측정되어 선택 되었고, 두 번째 특징은 두 눈 사이의 다른 명도로 측정되어 선택되었다.

얼굴 검출 단계에서 입력 이미지에 대하여 얼굴 검색 윈도우를 일정 크기로 멀티스케일 하였고, 일정 간격으로 입력 영상 위에서 쉬프트하면서 스캔을 수행하였고, 모든 서브 윈도우에 대해 스테이지 분류기

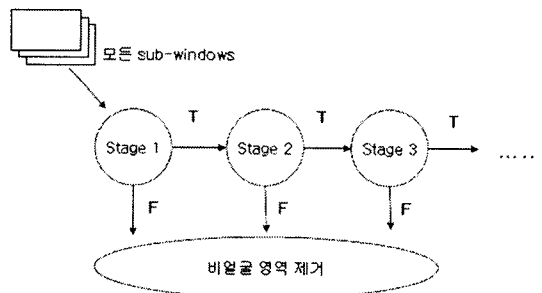


그림 5. 얼굴 검출을 위한 캐스케이드 구조

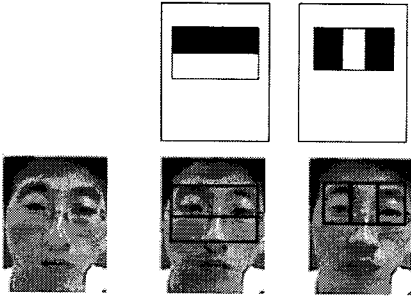


그림 6. AdaBoost 학습에 의해 선택된 특징 예

를 통해 얼굴 영역을 분류하였다. 초기 스테이지에서 많은 수의 배경영상이 제거되었고, 다음 스테이지는 이전 스테이지에서 제거되지 않은 배경영상을 제거하여 고려할 영상의 개수를 감소시켰다. 각 스테이지에서 배경영상이 제거되므로 연산량을 대폭 감소시켰다.

### 3. 주성분 분석 이용한 빠른 SVM 얼굴 영역 검증

#### 3.1 주성분 분석(Principal Component Analysis)

주성분 분석(Principal Component Analysis)은 고차원 입력 벡터를 저차원의 벡터로 표현하여 몇 개의 주성분 값으로 나타내어 주는 방식이다[14]. PCA를 이용하여 얼굴 영역 이미지의 특징 파라미터를 추출하는 것은 전체 영상의 데이터를 데이터의 분산이 큰 몇 개의 고유얼굴(EigenFaces)에 대한 축으로 선행 투사시켜 데이터의 차원을 줄이는 것이다. 여기에서 고유얼굴은 원래의 얼굴 영역 이미지에 일치하는 공분산 행렬(Covariance Matrix)의 고유벡터(Eigenvectors)를 나타낸다. 그림 8은 얼굴 영역 이미지의 특징 추출을 위한 주성분 분석에 대한 과정이 나타나 있다.

얼굴 이미지의 학습 집합( $X$ )를  $M$  개의 벡터로 표현하면 (7)과 같고, 각 벡터는  $N$ 개의 픽셀을 갖는다. 즉, 벡터  $x$ 는 얼굴 이미지의 전체 픽셀  $N$ 개를 갖는 벡터를 나타낸다.

$$X = \{x_1, x_2, \dots, x_M\} \tag{7}$$

$M$ 개의 얼굴 이미지의 평균벡터  $\Psi$ 는 (8)과 같고, 얼굴 이미지의 벡터 ( $x_i$ )와 얼굴 이미지의 평균벡터

( $\psi$ ) 사이의 차이는 (9)와 같이 구할 수 있다.

$$\Psi = \frac{1}{M} \left[ \sum_{i=1}^M x_i \right] \tag{8}$$

$$\Phi_i = x_i - \Psi, \quad i = 1, \dots, M \tag{9}$$

(10)을 이용하여 공분산 행렬 ( $C$ )을 구한다.

$$C = \frac{1}{M} \left[ \sum_{n=1}^M \Phi_n \Phi_n^T \right] \tag{10}$$

(10)에서 행렬 ( $C$ )의 고유값 ( $\lambda$ )과 고유벡터 ( $e$ )는 (11)의 방정식 해를 찾음으로써 구한다.

$$Ce = \lambda e \quad (e \neq 0) \tag{11}$$

계산된 고유벡터들의 행렬을  $A$ 라고 하고, 다음 (12)를 통해서 얼굴 이미지 벡터  $x$ 의 주성분 값 ( $y$ )를 구한다.

$$y = A(x - \Psi) \tag{12}$$

(12)를 이용하면 원래 얼굴 이미지 ( $x$ )로 복원하는 (13)을 재구성할 수 있다.

$$x = A^T y + \Psi \tag{13}$$

본 논문에서는 행렬  $A$ 의 고유벡터들을 모두 사용하지 않고 상위 몇 개( $k$ )를 고유벡터로 사용하였다. 이를 다시 표현하면 (14), (15)와 같다.

$$y = A_K(x - \Psi) \tag{14}$$

$$x = A_y^T + \Psi_K \tag{15}$$

본 실험에서는 고유얼굴(EigenFaces)을 인식할 데이터의 각 프레임에 사영(projection)시켜 얼굴 특징 계수를 추출하였다.

#### 3.2 서포트벡터머신(Support Vector Machine : SVM)

SVM은 1995 Vapnik[9]에 의해 이진패턴 분류 문제를 해결하기 위해 제안된 학습방법으로 각 범주의 대표적인 특성을 갖는 벡터를 이용하여 최적의 하이퍼플레인(Hyperplane)을 찾아서 두 범주를 분류하는 기법이다. SVM은 인식 성능과 처리 속도 성능이

매우 뛰어나 최근에 얼굴인식, 문자인식, 문서 범주화 등 다양한 분야에 적용되고 있다[8-10].

그림 7(a)에서와 같이 유클리드 공간에서 두 개의 범주를 갖는 N개의 객체가 p차원에 존재할 때, 두 범주를 분류하는 하이퍼분리면의 무수히 많이 존재하게 되고, 두 범주를 가장 효과적으로 나눌 수 있는 최적의 하이퍼분리면이 존재하게 된다.

그림 7(b)의  $X_1, X_2$ 와 같이 하이퍼분리면으로부터 가장 가까이에 위치한 데이터를 서포트벡터(Support Vector:SV)라 하고 하이퍼분리면에서 SV의 위치까지의 최소 거리를 마진(Margin)이라한다. 최적의 하이퍼분리면(Optimal Separating Hyperplane: OSH)은 마진을 가능한 최대로 하여 두 범주 간의 구분을 최적화한다.

그림 7(b)에서 하이퍼분리면들을 벡터의 내적으로 표현하면 (16)과 같고  $W$ 는 하이퍼분리면과 직교하는 가중치 벡터이고,  $b$ 는 기준값이다.

$$(W \cdot X) + b = 0 \quad W \in R^n, \quad b \in R \quad (16)$$

SVM의 학습 단계는 하이퍼분리면들 중에서 최적 하이퍼분리면을 찾는 과정이고, 최적의 하이퍼분리면은 마진  $2/\|W\|$ 가 최대가 되도록 하며,  $\|W\|/2$ 을 최소화하는 문제가 된다. SVM의 테스트 단계는 임의의 n차원 벡터의 입력 객체  $X_i \in R^n$ 을 두 개의 범주  $y \in \{+1, -1\}$ 로 예측하는 과정이고, 학습을 통해 계산된 최적 하이퍼분리면에 응답하는 분류 결정 함수  $f(X)$ 는 (17)과 같다.

$$f(X) = \sin((W \cdot X_i) + b) \quad (17)$$

선형 분리가 가능한 경우에는 (18)과 같은 최적화 문제가 된다.

$$\begin{aligned} \max \frac{2}{\|W\|} \Rightarrow \min \frac{\|W\|}{2} \\ \text{subject to } y_i(X_i \cdot W + b) - 1 \geq 0 \quad \forall i \end{aligned} \quad (18)$$

최적화 문제에서 함수의 최소 또는 최대 문제를 라그랑지(Lagrange) 함수로 유도하면 (19)와 같은 원문제(Primal)가 된다.

$$\begin{aligned} \min L_p = \frac{1}{2} \|W\|^2 - \sum_{i=1}^N \alpha_i [y_i(W \cdot X_i + b) - 1] \\ \text{subject to } \alpha_i \geq 0, \quad i = 1, \dots, N \end{aligned} \quad (19)$$

(19)에 대해서 울프 쌍대문제(Wolfe Dual Problem)로 유도하고 KKT(Karush-Kuhn-Tucker)조건을 적용하면, (21)을 제약조건으로 하고 (20)을 최대화하는 최적화 문제가 된다[9,10].

$$\begin{aligned} \max L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j X_i \cdot X_j \quad (20) \\ \text{subject to } \sum_{i=1}^N \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C \end{aligned} \quad (21)$$

(20), (21)에서 라그랑지 계수  $\alpha_i$ 는 2차 프로그래밍 문제(Quadratic Programming:QP)의 해[11]이며,  $W = \sum_{i=1}^N \alpha_i y_i X_i$  이고,  $C$ 는 마진 내의 객체를 허용하기

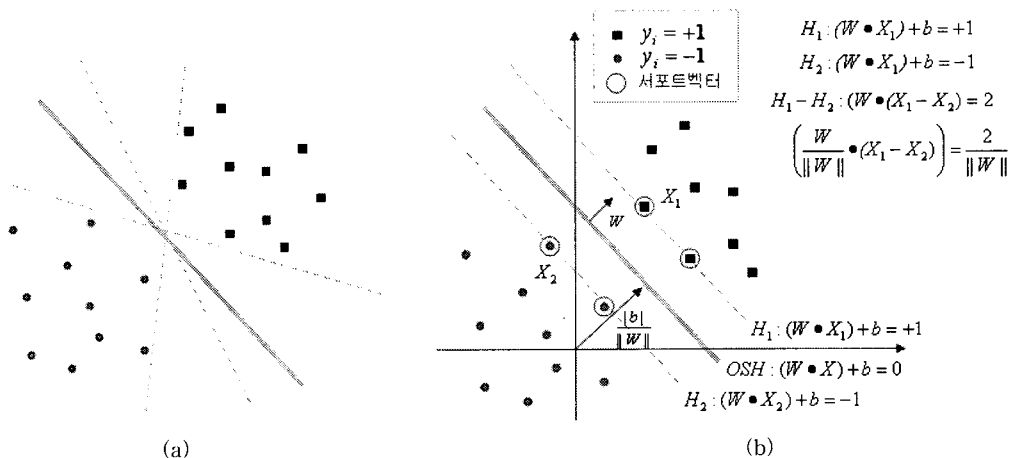


그림 7. 하이퍼분리면: (a) 하이퍼분리면, (b) 최적 하이퍼분리면.

위한 페널티 값이다.  $0 < \alpha_i < C$ 이면 서포트벡터(SV)가 되고,  $\alpha_i = C$ 이면 마진에 위치하고,  $\alpha_i = 0$ 이면 그 밖에 위치한다.

비선형 SVM은 선형 분리가 불가능한 경우에 입력 공간을 분리하는 비선형 최적 결정면을 이용하기 위해 두 벡터의 내적을 나타내기 위한 커널 함수로 입력 공간의 내적을 특징 공간의 내적으로 변형시켜 고차원으로 투영하여 계산을 용이하게 한다.

비선형 SVM 문제는 입력 데이터 공간을 커널 함수를 이용하여 보다 높은 특징 공간으로 맵핑하여 보다 높은 차원의 유클리드 공간으로 학습 데이터를 투영한다.

$$f(X_i) = \phi(X_i) \cdot W + b \tag{22}$$

(22)는 하이퍼분리면의 일반식이고  $\phi(X)$ 는  $X$ 의 기저함수로서 보다 높은 차원의 맵핑을 수행하는 함수이다. (23)과 같이 두 벡터의 내적을 나타내기 위한 커널 함수  $K(X, X_i)$ 을 이용하여 투영시켜서 선형 분류문제로 해결한다.

$$K(X, X_i) = \phi(X) \cdot \phi(X_i) \tag{23}$$

맵핑 함수를 사용한 라그랑지 듀얼 문제는 (24), (25)와 같다.

$$\max L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \phi(X_i) \cdot \phi(X_j) \tag{24}$$

$$\text{subject to } \sum_{i=1}^N \alpha_i y_i = 0, 0 \leq \alpha \leq C \tag{25}$$

일반적으로 사용되는 커널 함수는 (26)과 같다.

$$\text{Polynomial : } K(X, X_i) = (X \cdot X_i + 1)^d$$

$$\text{Gaussian : } K(X, X_i) = \exp\left(-\frac{\|X - X_i\|^2}{2\sigma^2}\right)$$

$$\text{Sigmoidal : } K(X, X_i) = \tanh(kX \cdot X_i - \delta) \tag{26}$$

### 3.3 주성분 분석을 이용한 얼굴 특징 추출과 SVM을 이용한 얼굴 검증

얼굴 이미지에서 PCA의 사용은 얼굴의 정보를 적게 가지고 있는 데이터를 제거하여도 얼굴의 표현에 영향이 적다는 점을 이용했다. PCA는 얼굴들에 대한 고유얼굴을 찾는 과정이므로 얼굴들만 가지고 학습을 수행하였고, SVM은 이진 분류기로서 얼굴, 비얼굴 각각을 PCA 변환을 통해 구해진 고유얼굴에 투영하여 얼굴(+1)클래스와 비얼굴(-1)로 하여 학습을 수행하였다.

그림 8에서 PCA 변환 후 저차원 데이터는 원 얼굴 이미지를 구분 할 수 있는 많은 정보를 가지고 있으므로 본 논문에서는 SVM 검증을 위한 특징 계수를 PCA 변환을 이용하여 빠른 실시간 얼굴 검증 시스템을 구현 하였다. 검증 과정에서 얼굴 영상이 입력으로 들어오면, 얼굴 영상으로 구성된 고유얼굴들과 거리가 가까우므로 얼굴로 검출하게 되고 비얼굴 영상이 입력으로 들어오면 고유얼굴들과 멀어 지므로 비얼굴로 검출하게 된다.

본 논문에서 제안한 PCA와 SVM을 이용한 얼굴 검증을 실험하기 위해 순수 SVM 방법과 SVM-

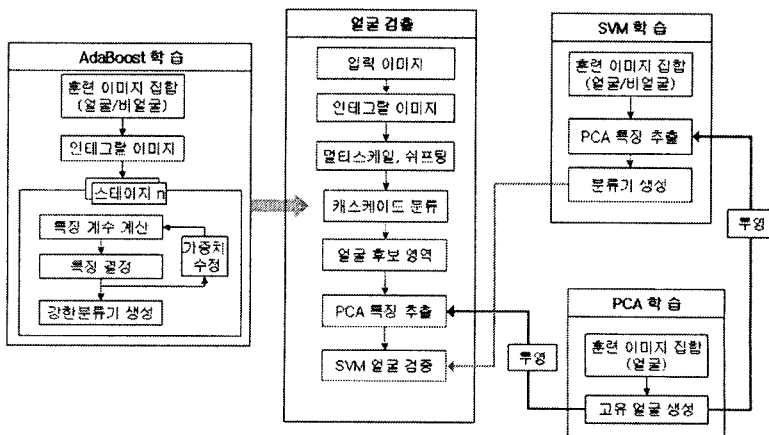


그림 8. 얼굴 영역 검증 구성도

PCA 복합 알고리즘 방법을 구축하였다. 먼저, 순수 SVM의 학습을 위해 훈련이미지를 19×19=361 크기로 정규화하고 마스크를 적용하여 훈련데이터를 만들었고, 그림 9에서 정규화 된 얼굴 이미지는 약간 배경과 헤어가 차지하고 있으므로 마스크를 적용하여 비얼굴 영역을 제거와 함께 훈련데이터 벡터를 301로 축소하였다. 이러한 훈련데이터에 비선형 SVM을 적용하였고, 매개변수인 오류 허용치를 200을 설정하여 학습을 수행하였다.

두 번째, 훈련 이미지를 PCA 변환을 통해 고유얼굴과의 거리 정보를 이용하였다. 고유얼굴과의 거리가 작을 수록 얼굴에 근접한 특징계수가 된다. PCA 변환으로 고차원 데이터를 저차원의 데이터로 축소하고 비선형 SVM의 학습 데이터를 사용하였다.

순수 SVM 방법과 SVM+PCA 복합 방법의 학습 결과로 각각 서포트벡터와 서포트벡터에 해당하는 (19)에서의 나타난  $\alpha$  값을 획득하였다. 이러한  $\alpha$  값은 SVM의 이차계획 최적화를 통해 획득할 수 있다. 본 논문에서는 SVM의 2차 계획(QP) 최적화 문제를 해결하기 위해 학습 알고리즘인 John C. Platt의 SMO(Sequential Minimal Optimization)을 이용하여 구현하였다[11]. SMO는 2차 계획 문제를 작은 2

차 계획 문제들로 분해해서 분석(Analytic)으로 해결하고, 아주 적은 코드로 구현이 가능하고, 빠른 수렴 성능이 보고되었다.

#### 4. 팬-틸트 웹카메라를 이용한 자동 추적형 얼굴 검출 시스템

본 논문에서는 팬(Pan)-틸트(Tilt) 기능을 가진 저가형 웹카메라인 스마일캠을 이용하여 사람의 얼굴을 추적하였다. 스마일캠은 CCD 1/3인치 이미지 센서, NTSC 270K, PAL 290K, 팬  $\pm 90^\circ$  (90°/sec), 틸트  $\pm 30^\circ$  (90°/sec)의 사양을 보유하고 있다. 스마일캠의 이동 제어를 하기위해 RS232 통신을 이용하였다. 시리얼 포트를 제어하기 위한 통신 사양으로 Baud Rate는 19200bps, Data bit는 8bits, Stop bit는 1bit, Parity bit는 none으로 설정하였다. 얼굴 후보 영역에 대한 검증을 통해 얼굴로 판명된 영역 정보인 위치, 크기를 이용하여 팬-틸트를 수행하였다. 카메라의 얼굴 추적에 대한 이동을 위해 획득된 영역에 대한 중심점을 찾은 다음, 중심점의 위치 정보를 이용하여 상하좌우 이동하여 이미지 중심에 얼굴을 위치하도록 하였다.

$$\alpha(X, Y) = \left( \frac{x+w}{2}, \frac{y+h}{2} \right) \tag{27}$$

그림 10(a)에서와 같이 얼굴이 검출되면 (30)과 같이 검출된 이미지 내의 중심 좌표  $\alpha(X, Y)$ 는 얼굴 영역의 중심 좌표  $\left( \frac{x+w}{2}, \frac{y+h}{2} \right)$ 로 변경되면서 그림 10(b)와 같이 카메라가 팬-틸트를 수행한다. 그림

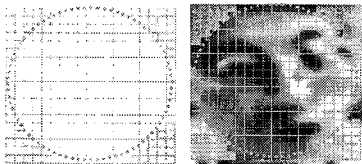


그림 9. SVM의 특징계수에 대한 마스크

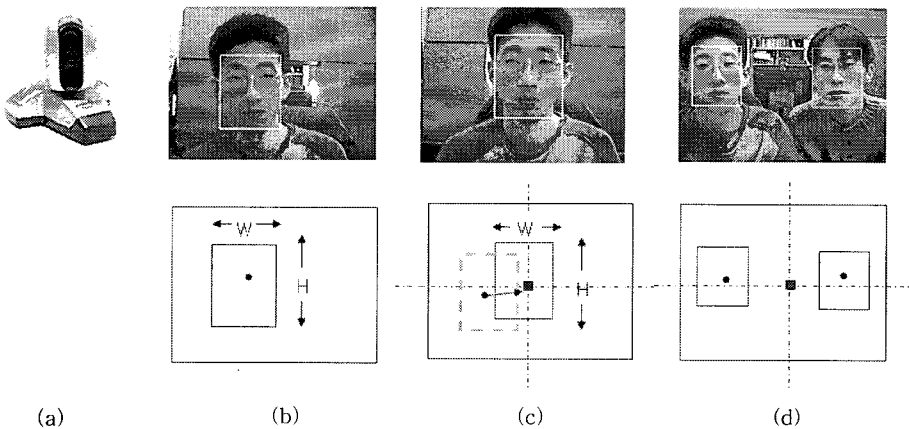


그림 10. 팬-틸트 실시간 얼굴 추적: (a) 팬-틸트 웹카메라, (b) 얼굴 추적 전, (c) 얼굴 추적 후, (d) 다중 얼굴 추적.



10(d)와 같이 다중 얼굴이 검출된 경우에는 획득된 얼굴들의 중심으로 이동하였다.

팬-틸트 웹카메라의 부드러운 이동을 위해 실시간 얼굴 추적을 위해 평균 초당 5프레임 이상의 성능이 요구되었다. 본 논문에서 제안된 얼굴 추적의 시간은 AdaBoost의 평균 20ms와 얼굴 검증 PCA+SVM의 평균 0.9ms이었다. 따라서 얼굴 검출을 위한 총 평균 시간은 20.9ms로 웹카메라의 팬-틸트의 부드러운 움직임을 보였다.

### 5. 실험 및 결과

본 논문에서 제안하는 얼굴 검출 시스템의 실행 화면의 예는 그림 11에 나타나 있고, 실험 환경은 표 1과 같다.

AdaBoost의 학습을 위해 MIT 얼굴 데이터베이스[12]를 이용하였다. 그림 12의 MIT 얼굴 DB는 훈련용 총 6,977개 이미지로 2,429개 얼굴과 4,548개 비얼굴로 구성되고, 시험용으로 총 24,045개 이미지로 472개의 얼굴과 23,573개 비얼굴로 구성되어 있다.

학습 이미지 크기는 24×24로 정규화 하였고 24개의 스테이지의 캐스케이드 구조로 처음 5개의 스테

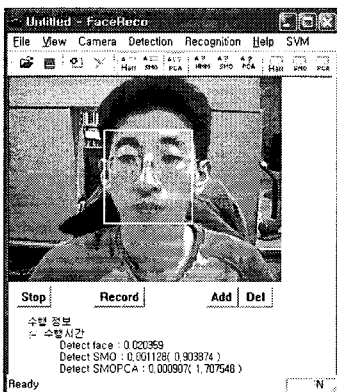
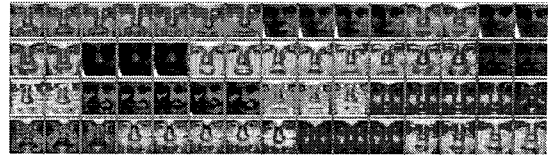


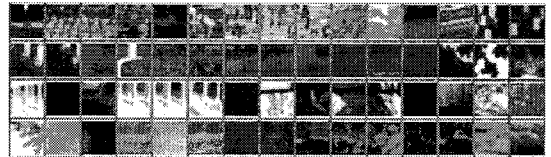
그림 11 실행 화면 예

표 1. 실험 환경

구분	환경	비 고
운영체제	Windows XP	
CPU	Intel Pentium IV 3.0GHz	
프로그램 언어	MS Visual C++ 6.0	
입력 장치	웹카메라 4종류	팬-틸트 기능(1개)



(a)



(b)

그림 12. MIT 얼굴 DB: (a) MIT 훈련용 얼굴 DB 샘플, (b) MIT 시험용 비얼굴 DB 샘플.

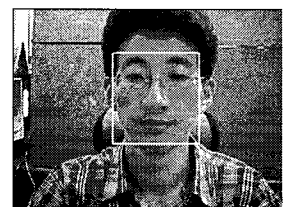
이지의 특징계수의 수는 1, 10, 25, 25, 50로 각각 설정하였고 나머지 스테이지는 점점 증가시켜 최종 전체 특징계수 6061개를 사용하였다.

AdaBoost 테스트 단계에서는 웹카메라의 영상 320×240 크기의 이미지를 입력으로 하여 탐색윈도우를 이용하여 멀티스케일과 쉬프트하면서 얼굴 후보영역을 검출하였다. 멀티스케일에서 입력이미지를 멀티스케일하지 않고 검색 윈도우에 대해서 1.25 스케일 값을 설정하여 수행하였다. 쉬프트 단계에서 쉬프트 간격은 2픽셀로 설정하여 수행하였고, 검출된 후보영역들 중에 각 영역 내부에 존재하는 영역은 제거하였다. 그림 13은 실시간 얼굴 영역 검출 예이다.

PCA의 학습은 그림 14에 나타난 BioID(HumanScan) 얼굴 데이터베이스[13]를 추가하여 수행하였다. BioID 얼굴 데이터베이스는 1521개의 그레이 스케일 이미지로 구성되어 있고 384×286 픽셀의 크기와 각각 이미지는 23명의 정면(거의 정면)과 다양한 포즈로 구성되어 있다.



(a)



(b)

그림 13. 실시간 얼굴 영역 검출 예: (a) 스틸이미지, (b) 웹카메라.

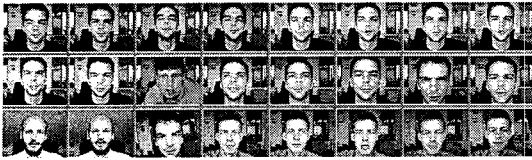


그림 14. BioID(HumanScan) 얼굴 데이터베이스

PCA의 학습은 얼굴들의 픽셀 값에 대한 공분산을 이용하여 고유얼굴을 도출하는 과정으로 비얼굴 데이터는 필요하지 않는다. 따라서 BioID 이미지의 비얼굴 영역은 제거하고 얼굴영역에 대해서만 선택하였고, 선택된 얼굴 영역은 눈썹, 눈, 코, 입을 필수로 포함하고 있는 1521개 중에서 1443개를 선택하여 학습하였다.

PCA 학습을 위한 훈련 얼굴 이미지는 그림 15와 같이 32×32 크기로 정규화 하여 1024 크기의 훈련 데이터 벡터로 구성 하였다. 따라서 총 1024(이미지 크기)×1443(프레임 수) 크기 행렬을 이용하여 평균얼굴을 계산하고, 공분산 행렬을 이용하여 고유벡터와 고유값을 계산하였다. 그림 16은 PCA 학습을 통해 획득된 고유얼굴이다. 총 1024개의 고유벡터와 고유값이 추출되었고, 첫 번째 이미지는 평균 이미지이

다. 또한 데이터의 정보량을 줄이기 위해 고유값의 크기에 따라 내림차순으로 고유벡터 정렬하여 상위 에 존재하는 데이터를 특징 계수로 사용하였다.

SVM 학습 단계에서는 BioID 얼굴 정규화 데이터와 MIT CBCL의 훈련용 얼굴, 비얼굴 데이터베이스를 이용하였다. 각 얼굴, 비얼굴 이미지들은 PCA 고유얼굴에 투영시켜 특징 계수를 생성하였고, 생성된 특징 계수는 SVM의 학습용 훈련데이터로 이용하였다. 얼굴 후보 영역이 얼굴 인지 아닌지를 판단하기 위해서 비선형 SVM 분류기를 이용하였고, 2차 다항식 커널과 페널티 C=200으로 설정하였다. 본 논문에서 적용한 PCA와 SVM의 복합 알고리즘의 성능을 테스트하기 위해 순수 SVM의 학습과 검증도 함께 실험하였다. 그림 17은 AdaBoost 알고리즘에서 단독으로 사용할 경우에 오검출한 얼굴 영역의 예로써 그림 17의 예에서 보이는 것과 같이 오검출의 원인으로서는 이미지내에서 얼굴과 비슷한 형태가 나온 경우에 발생하였다.

PCA 학습 단계에서 32×32 크기를 입력으로 하여 1024×1024 공분산 행렬을 만들어 사용 하였고, PCA에 학습에 의해 생성된 고유얼굴의 수는 1024개이다. 고유얼굴들을 고유값이 가장 큰 순으로 내림차순 정



그림 15. PCA 학습을 위한 BioID의 정규화 얼굴 영역 이미지



그림 16. PCA 평균얼굴과 고유얼굴



그림 17. AdaBoost의 얼굴 영역 오검출 예

렬하여 표 2와 같이 설정하여 각각을 실험하였다. 그림 18의 실험 결과를 보면 PCA 특징 계수 60개를 사용하였을 때 가장 좋은 결과를 보였다. 그림 18에서의 검출율은 이미지 내의 존재하는 모든 얼굴을 정확하게 찾은 경우를 적용하였고, 오검출율은 배경을 얼굴로 잘 못 판단하는 경우를 적용하였다. 또한 이미지 내의 존재하는 얼굴을 검출하지 못한 경우를 검출실패로 정의를 하였다.

본 논문에서 제안하는 얼굴 검출 방법은 표 3의 실험 결과와 같이 AdaBoost의 오인식된 얼굴 영역은 PCA(60개)+SVM의 얼굴 영역 검증의 수행을 통해 검출률과 검출시간의 큰 변동 없이 얼굴이 아닌 배경을 얼굴로 잘 못 판단하는 오검출률을 낮출 수 있다.

### 6. 결론

본 논문에서는 실시간 다중 얼굴 검출 시스템을 구현하였다. 본 논문에서 사용된 AdaBoost, PCA,

SVM은 모두 통계적 학습 기반 알고리즘이다. 학습 기반 알고리즘은 학습 과정에서 사용자가 직접 학습 샘플데이터, 매개변수(Parameter) 값, 커널 함수를 미리 결정해야 한다. 그러므로 최적의 성능을 구현하기 위해 다양한 데이터를 학습 알고리즘에 이용하여 실험에 의한 최적의 값을 선택하였다. AdaBoost 분류 방법은 간단한 Haar-like 특징들을 인테그랄 이미지 이용하여 연산을 빠르게 하였고, 캐스케이드 구조는 스테이지별로 특징 수를 다르게 하여 초기 스테이지에서 많은 배경을 제거하여 계산 속도를 빠르게 하였다. 또한 검출된 얼굴 영역에 대한 특징계수를 PCA를 이용하여 효과적으로 축소하여 SVM을 이용하여 얼굴 후보 영역 검증 과정을 추가하여 얼굴 검출의 정확도를 향상시켰다. 기존 얼굴 검출 방법보다 계산 비용을 줄였기 때문에 실시간 다중 얼굴 검출 성능이 향상되었다. 본 논문의 실험결과에서 나타나듯이 움직이는 팬-틸트형 웹카메라에 적용이 가능한 빠른 검출 성능을 보였고, 움직이는 얼굴을 추적하므로 얼굴 검출의 성능을 향상 시켰다.

표 2. PCA 계수 크기별 검출률

구분	PCA 100	PCA 90	PCA 80	PCA 70	PCA 60	PCA 50	PCA 40	PCA 30	PCA 25	PCA 20	PCA 15
검출률(%)	95.9	96.2	96.0	96.0	96.0	94.3	90.5	88.2	86.4	86.1	81.8
오검출률(%)	0.21	0.19	0.22	0.22	0.22	0.26	0.54	1.17	1.46	1.92	2.48
검출시간(ms)	24.3	23.5	22.6	21.8	20.9	20.1	19.3	18.5	17.9	17.4	16.6

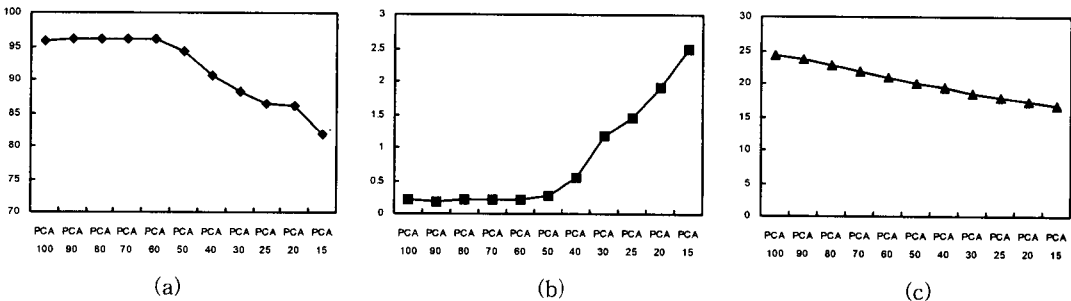


그림 18. PCA 특징 계수별 얼굴 검출 실험 결과: (a) 얼굴 검출률, (b) 얼굴 오검출률, (c) 얼굴 검출 시간.

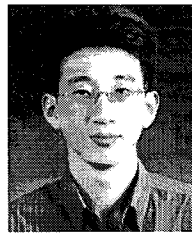
표 3. 실시간 얼굴 검출 실험 결과

구분	AdaBoost	AdaBoost+SVM	AdaBoost+CA(60)+SVM
검출률(%)	96.1	95.9	96.0
오검출률(%)	3.80	0.52	0.22
검출시간(ms)	20.0	22.2	20.9

참 고 문 헌

- [ 1 ] Ming-Hsuan Yang, Kriegman, D.J, and Ahuja N., "Detecting face in images : a survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 24, Issue:1, pp. 34-58, 2002.
- [ 2 ] G.Yang and T.S. Huang, "Human Face Detection in a Complex Background," *Pattern Recognition*, Vol. 27, No. 1, pp. 53-63, 1994.
- [ 3 ] Haiyuan WU and Qian CHEN, "Detecting Human Face in Color Images," *Proc. of IEEE*, pp. 2232-2236, 1996.
- [ 4 ] R.Brunelli and T.Poggio, "Face Recognition: Features versus Templates," *IEEE Trans. PAMI.*, Vol. 15 pp. 1042-1052, 1993.
- [ 5 ] Viola, P and Jones, M, "Rapid object detection using a boosted cascade of simple features," *Computer Vision and Pattern Recognition 2001(CVPR 2001)*, Vol. 1, pp. 511-518, Dec. 2001.
- [ 6 ] Lienhart, R and Maydt, J, "An extended set of Haar-like features for rapid object detection," *International Conference on Image Processing 2002(ICIP2002)*, Vol. 1, pp. 900-903, Sept. 2002.
- [ 7 ] Rainer Lienhart, Alexander Kuranov, and Vadim Pisarevsky, "Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection," *DAGM'03, 25th Pattern Recognition Symposium*, Madgeburg, Germany, pp. 297-304, Sep. 2003.
- [ 8 ] V. Vapnik, "The Nature of Statistical Learning Theory," *Springer-verlag*, New York, 1995.
- [ 9 ] C.J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," in *Data Mining and Knowledge Discovery*, Vol. 2, No. 2, pp. 121-167, 1998.
- [10] E. Osuna, R. Freund, and F. Girosi, "Training Support Vector Machines: An application to face detection," *Proceeding IEEE. CVPR*, pp. 130-136, 1997.
- [11] Platt, J.C., "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines," Microsoft Research Technical Report MSR-TR-98-14, 1998.
- [12] Center for Biological and Computational Learning at MIT and MIT, "CBCL DATASETS," <http://cbcl.mit.edu/cbcl/software-datasets>, 2004.
- [13] HumanScan GmbH, "The BioID Face Database," <http://www.bioid.com>, 2004.
- [14] Ming-Hsuan Yang, "Kernel Eigenfaces vs. Kernel Fisherfaces: Face Recognition Using Kernel Methods," *Automatic Face and Gesture Recognition, Proceedings. Fourth IEEE International conference on*, pp. 208-213, 2002.

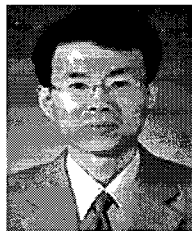
이 호 근



2002년 2월 원광대학교 컴퓨터 및 정보통신공학부(공학사)  
 2005년 2월 원광대학교 컴퓨터 공학과(공학석사)  
 2005년 11월 ~ 현재 아이디에스 IS평가기술구룹 근무

관심분야: 영상인식, 컴퓨터비전, 영상처리

정 성 태



1987년 2월 서울대학교 컴퓨터 공학과(공학사)  
 1989년 2월 서울대학교 컴퓨터 공학과(공학석사)  
 1994년 8월 서울대학교 컴퓨터 공학과(공학박사)  
 1995년 3월 ~ 현재 원광대학교

전기전자및정보공학부 교수

관심분야: 영상인식, 영상처리, 컴퓨터 그래픽스