

모바일 환경에서 Haar-Like Features와 PCA를 이용한 실시간 얼굴 인증 시스템*

김정철** · 허범근*** · 신나라*** · 홍기천****

Implementation of Realtime Face Recognition System using Haar-Like Features and PCA in Mobile Environment

Kim, Jung Chul · Heo, Bum Geun · Shin, Na Ra · Hong, Ki Cheon

〈Abstract〉

Recently, large amount of information in IDS(Intrusion Detection System) can be unmanageable and also be mixed with false prediction error. In this paper, we propose a data mining methodology for IDS, which contains uncertainty based on training process and post-processing analysis additionally. Our system is trained to classify the existing attack for misuse detection, to detect the new attack pattern for anomaly detection, and to define border patter between attack and normal pattern. In experimental results show that our approach improve the performance against existing attacks and new attacks, from 0.62 to 0.84 about 35%.

Key Words : Intrusion Detection, Data Mining, Classification, Fault Tolerance System, SVM

I. 서론

보안이 중요시 여겨지는 21세기에는 생체 인식 관련 기술들이 활발하게 연구되어지고 있다. 생체인식 중에 시각인식 기술의 연구도 활발하게 진행 되고 있으며, 시각 중 얼굴을 이용한 얼굴인식 및 동작 인식 얼굴의 표

정 인식 기술 등 다양한 기술들이 사람과 컴퓨터의 상호 작용의 수준을 향상 시키려는 연구가 활발해 지고 있다 [1]. 시각인식 기술에 있어서 실시간으로 영상을 처리하는 기술은 무엇보다도 중요하며, 단순 정지 영상 이미지를 처리하는 기술 보다는 실시간 시스템에서 영상을 처리 하는 기술이 시각 인식 관련 응용프로그램들이 많이 있으며, 그 활용 범위도 넓다. 본 논문에서는 이러한 시각인식 관련 응용프로그램 중 하나로 카메라를 이용한 실시간 얼굴 검출 및 시스템을 구현하였다.

* 본 연구는 경기도의 경기도지역협력연구센터사업의 일환으로 수행하였음[(GRRR 수원2009-B4), 지능형 감시를 위한 이벤트 추출 시스템 개발].

** (주)하이트론시스템즈(제1저자)

*** 수원대학교 정보통신공학과 석사과정

**** 수원대학교 정보통신공학과 교수(교신저자)

II. 구현환경

본 논문에서 얼굴 인증 시스템은 임베디드 시스템 상에서 구현을 하였으며, OS는 RedHat 9.0을 사용했다. 또한 구현 보드는 HYBUS사의 X-Hyper255B TKU3로 구현을 하였다. 언어는 c++를 사용했으며, QT 라이브러리를 이용하여 GUI를 구현 하였다[2]. 얼굴 검출 및 인증 알고리즘은 Haar Like Feature 와 PCA를 이용하였다.

2.1 임베디드 리눅스 시스템의 사양

본 시스템에서 구현 한 얼굴 검출 및 얼굴 인증 시스템의 타겟보드는 HYBUS사의X-Hyper255B TKUIII이다 [3]. X-Hyper255B TKUIII의 하드웨어적 사양은 <표 1>과 같다. 컴파일러는 하이버스사에서 제공해준 툴 체인을 사용함으로써, 직접 컴파일러를 다운받아 업그레이드 및 패치를 해야 하는 수고를 덜었다. LCD는 6.4인치 크기를 가지고 있으며, 이 프레임 버퍼를 이용하기 위한 GUI는 QT를 사용했다. 리눅스 커널은 2.4.18 버전을 선택하였다.

<표 1> X-Hyper255B TKUIII의 하드웨어 사양

Item	Description
컴파일러	GCC(GNU Compiler Collection) 2.95.3
Device Drivers Frame	Buffer (6.4" TFT LCD) Touch Screen UART (Console) USB Host Sound (Microphone / Speaker) PCMCIA
OS	LINUX 2.4.18

III. 얼굴 검출

AdaBoost 학습 알고리즘에 의해서 얼굴과 눈을 잘 표현하게 할 수 있도록 프로토타입을 정하는데, 이것을 가

지고 얼굴 검출 알고리즘인 Haar-like feature에서 주로 사용하므로 효율적인 얼굴 검출이 가능하다[4]. 이런 알고리즘을 얼굴 검출 시스템에 적용한 것으로 단순 합 이미지를 이용하여 계산을 하므로 그 어떤 알고리즘보다 연산 속도에 있어서 빠르다는 장점을 가지고 있다[5-6].

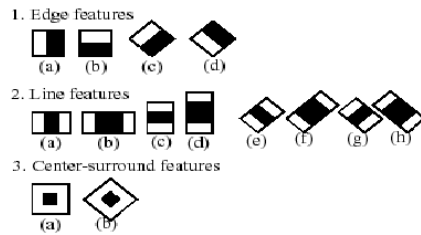
3.1 Haar-like feature의 원리

Haar-like feature는 얼굴을 검출하는데 있어 픽셀을 이용한 연산이 아닌 특징을 이용하여 각 영역 안에 있는 픽셀들의 값을 더하여 영역 합을 구하고 그 값들에 가중치(weight)를 곱한 합만을 계산한다는 것으로 동영상에서의 얼굴 검출에 용이하다.

<그림 1>은 Haar-like feature의 prototype들로서 윈도우에서 특징에 맞게 위치가 변하기 때문에 인식할 영상의 따라 많은 특징 값을 나타내는 것이 가능해진다.



(A) A set of basic Haar-like features



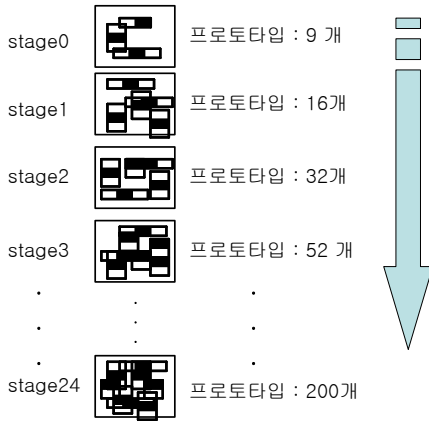
(B) A set of extended Haar-like features

<그림 1> Haar-like feature의 프로토타입

3.2 AdaBoost의 원리

본 논문에서는 Boost 알고리즘 중에서 단순하고 효율적인 AdaBoost 학습 알고리즘을 사용함으로써 얼굴과 얼굴이 아닌 영역간의 구별과 얼굴로 판단할 확률을 높

이다. 이 알고리즘의 장점은 classifier하는 stage가 올라갈수록 각 Haar-like feature의 프로토타입들의 위치가 얼굴의 미세한 정보까지도 나타내는 장점을 가지고 있다. AdaBoost 학습 알고리즘으로 얻은 특징값들은 <그림 2>와 같이 단계(stage)별로 그룹화(classifier)한다

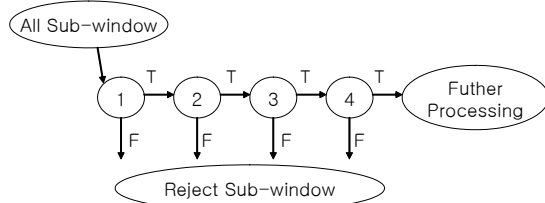


<그림 2> 프로토타입들의 증가 블록도

이러한 그룹화는 더 강력한 인식 알고리즘을 구현하기 위한 것으로 단계를 거듭할수록 전 단계보다 더 많은 수의 특징값을 만들어서 그룹화를 하며 본 논문에서는 얼굴을 25단계로 그룹화를 하여 실험하였다.

3.3 Haar-like feature를 이용한 얼굴 검출 방법

AdaBoost 학습 알고리즘으로 얻은 특징 값들은 <그림 3>과 같이 단계(stage) 그룹화(classifier)한다.



<그림 3> AdaBoost를 이용한 각 단계별 그룹화

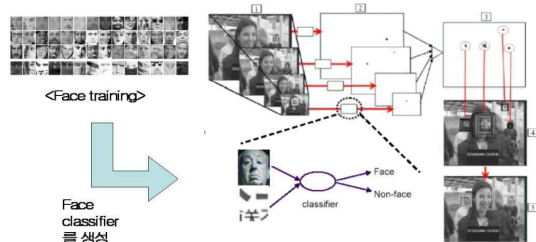
이러한 Haar-like feature들의 그룹화는 단계를 거듭할수록 전 단계보다 더 많은 수의 특징 값을 만들어서 더욱더 강력한 얼굴 인식 알고리즘을 구현해 준다.

그 다음 AdaBoost 학습 알고리즘을 통해 선택된 Haar-like feature를 그룹화 하여 저장한다. 첫 번째 단계의 그룹에서는 가장 적은 9개의 Haar-like feature 가 들어간다. 각 단계는 총 25단계로 그룹화 되고 이때에는 200개의 Haar-like feature가 들어가게 된다. 이렇게 그룹화한 데이터들은 XML파일로 저장이 되어 얼굴 인식에 있어서 hidden cascade로 사용이 된다[7].

본 논문에서는 24*24크기의 윈도우로 정해졌고 높은 단계로 올라갈수록 낮은 단계보다 사용하게 되는 Haar-like feature의 개수도 증가하고 각각의 프로토타입들도 얼굴 특징점의 세밀한 부분까지 위치하게 된다.

3.4 Haar를 이용한 얼굴 검출의 원리

<그림4>는 Haar-like feature를 이용한 얼굴 검출 원리를 간단하게 표현한 것이다. 즉 한 프레임의 영상에서 트레이닝된 얼굴의 그룹을 이용하여 얼굴 영역의 세밀한 정보를 얻기 위하여 영상을 피라미드 구조의 형식으로 줄여나가며 얼굴 영역을 결정을 하게 된다. 그러한 과정에서 얼굴 영역은 줄인 영상을 복원하는 과정에 여러 개의 후보영역이 생기게 되며 그 영역의 평균을 낸 한 영역을 출력으로 하게 된다[8].



<그림 4> Haar Face Detection의 흐름도

IV. 얼굴 인식

PCA(Principal Component Analysis)란 주성분 분석법이라 하며, 주로 다루기 힘든 고차원의 신호를 낮은 차원으로 줄여 다루기 쉽게 해주는 통계적 방법을 일컫는다.

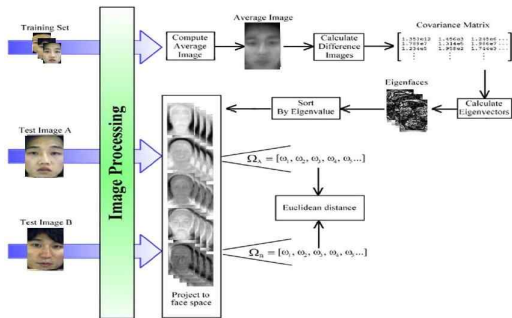
PCA를 이용하여 등록되는 영상들은 얼굴만을 위한 고유공간에서의 새 좌표계로 변환되어 저장되며 후에 인식 시에는 새로 들어온 영상들을 역시 얼굴만의 고유공간상의 좌표계로 변환하여 그 둘 사이의 떨어진 거리를 측정함으로써 등록된 얼굴과의 일치 여부를 결정하게 된다[9-10].

4.1 PCA를 이용한 얼굴 인식

고유얼굴을 부호화하여 즉 특징점들에 대해 데이터베이스를 저장한 후 이 데이터베이스와 비교 분석을 통하여 인증을 실시한다.

고유얼굴은 트레이닝 이미지 데이터들로부터 획득되어지는 covariance matrix 의 eigenvector들로 연산되어진다. 그 eigenvector들은 N*N차원의 이미지에서 covariance matrix의 orthonormal basis로부터 non-zero eigenvalues 값들에서 구해진다. 특히 각 이미지는 size N 벡터 저장되어진다.

아래 <그림 5>는 PCA 알고리즘의 흐름도이다.



<그림 5> PCA 알고리즘의 흐름도

식 (6-1)은 각 이미지들의 벡터들에 대한 정의를 나타낸다.

$$x^i = [x_1^i \dots x_N^i]^T \quad (6-1)$$

2D 벡터로 된 이미지를 고유얼굴로 매핑하기 위해 1D 벡터로 만들어야 하며 각 이미지들의 벡터들에 대해 평균을 구하는데 그 식은 식 (6-2)와 같다.

$$m = \frac{1}{p} \sum_{i=1}^p x^i \quad (6-2)$$

평균 데이터를 이용하여 각각의 이미지에 대한 차이 이미지를 생성하는 데 그 식은 (6-3)와 같다

$$\bar{x}^i = x^i - m \quad (6-3)$$

각각의 이미지에 대해 차이 이미지를 \bar{x}^i 로 정의를 할 때 이 차이 이미지의 조합은 식 (6-4)와 같은 식으로 나타내어지며, 그 데이터 매트릭스의 크기는 N*P(P는 이미지의 수)이다.

$$\bar{X} = [\bar{x}^1 \dots \bar{x}^P] \quad (6-4)$$

Covariance matrix는 각각 차이 이미지의 조합에 대한 벡터와 그 벡터에 대한 트랜스 포즈(T)를 한 것과 같다. 식 (6-5)는 이를 나타낸다.

$$\Omega = \bar{X} \bar{X}^T \quad (6-5)$$

4.2 고유얼굴을 이용한 인식 알고리즘

고유얼굴로의 투사(projection)를 통하여 인식할 이미지는 기본적으로 아래와 같은 3단계를 수행한다.

고유공간은 트레이닝 이미지들을 이용하여 생성되어야 한다.

트레이닝 이미지들을 고유공간으로 투사시켜야 한다.

테스트 이미지는 고유공간으로 투사시키고 테스트 이미지와 투사 된 트레이닝 이미지와 비교를 함으로써 인식되어 진다.

1) 고유공간 생성 알고리즘

아래와 같은 단계를 거쳐서 고유공간을 생성한다.

차 데이터(Center data) : 각각의 이미지들에 대해서 평균을 구해서 식 (6-3)과 같이 트레이닝 이미지들과의 차를 구한다. 평균 이미지는 열 벡터이다.

데이터 매트릭스(data matrix) 생성 : 트레이닝 이미지를 차 데이터로 연산 후 사이즈를 N*P의 데이터 매트릭스의 조합으로 만든다. 여기서 P는 트레이닝 이미지의 수이며 각 열은 식 (6-4)와 같이 하나의 이미지이다.

공분산 매트릭스(covariance matrix) 생성 : 데이터 매트릭스와 이 매트릭스의 트랜스포즈 한 것과 곱을 통하여 식 (6-5)와 같이 공분산 행렬은 생성이 된다.

고유값과 고유벡터(eigenvalues and eigenvectors) 생성 : 고유값과 그에 대응하는 고유벡터는 공분산 행렬을 이용하여 생성이 되어 지며 아래 식 (6-6)과 같다.

$$\Omega V = \Lambda V \tag{6-6}$$

여기서 V 는 eigenvalues Λ 에 대응되는 고유벡터들의 행렬이다.

고유벡터 순서(Order eigenvectors) : 고유값($\lambda_i \in \Lambda$)는 높은 값에서 낮은 값으로 정렬이 되며, 이에 대응하는 고유벡터($\lambda_i \in V$)는 고유값에 따라서 정렬이 된다. 단 영값을 가지지 않는 고유값에 대한 고유벡터들이다. 고유벡터에

대한 매트릭스는 고유공간 V 이다. 여기서 V 의 각 열은 고유벡터이다. 식 (6-7)은 이에 대한 공식을 나타낸다.

$$V = [V_1 \dots V_p] \tag{6-7}$$

2) 트레이닝 이미지 투사

각각의 차 트레이닝 이미지들을 고유공간으로 투사시킨다. 고유공간으로 투사시키기 위하여, 정렬 된 고유벡터들의 이미지들에 내적을 계산한다.

$$\tilde{x}^i = V^T X^i \tag{6-8}$$

내적 이미지와 첫 eigenvector는 새로운 벡터에서의 처음 값이다. 투사 된 이미지의 새로운 벡터는 고유값들 만큼 많은 값을 포함한다.

3) 이미지 인식 알고리즘

식 (6-9)는 인식할 이미지를 평균 이미지에서 뺀 후 차 데이터를 나타낸다.

$$\bar{y}^j = y^j - m \tag{6-9}$$

여기서 y 는 인식할 이미지이다.

그 다음에 (6-10)에서 보는 바와 같이 트레이닝 데이터를 가지고 정의한 고유공간으로 투영을 시킨다.

$$\tilde{y}^j = V^T \bar{y}^j \tag{6-10}$$

투사된 인식할 이미지는 모든 투사된 트레이닝 이미지와 비교를 한다. 인식할 이미지와 가장 가까운 값을 가지는 트레이닝 이미지는 인식 이미지가 된다. 이 측정은 norm 값을 이용하여 비교한다.

V. 실험 환경 및 고찰

5.1 실험 환경

본 논문의 실험에 사용된 환경은 다음과 같다. 먼저 데스크탑에서는 컴퓨터 Pentium(R)4-1. 80GHz와 Window XP를 사용하였다. 카메라는 Logitech카메라를 이용하였으며, 타겟 보드에서는 TKUIII와 ov511 카메라를 사용하였다. 영상 이미지는 bmp파일과 jpg파일을 사용하여 실험을 하였다. 또한 알고리즘의 구현은 window XP에서는 Visual C++을 사용하였고, 타겟 보드에서는 QT라이브러리를 이용하였다.

영상 이미지의 크기는 320×240이며, 건물 내부에서 실험을 시행하였으며, 얼굴검출 및 인식 알고리즘의 경우 그레이 레벨에서의 연산을 하기에 조명에 민감하여 미디언 필터 윈도우 이미지의 크기는 가장 적합한 81×81로 결정하였다.

5.2 Haar-like feature와 PCA에 대한 실험

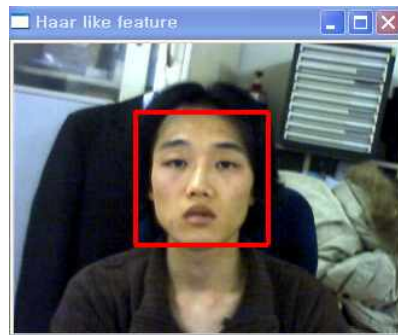
<그림 6>은 여러 포즈의 정지 영상 이미지에서 얼굴 검출이다. 이 영상 이미지에서 얼굴이 30정도 기울여 지거나 얼굴 영역이 너무 작아서 검출하기 힘든 경우를 제외 하고는 얼굴 검출을 하는 결과를 보여 준다.



<그림 6> 여러 포즈에서의 얼굴 검출

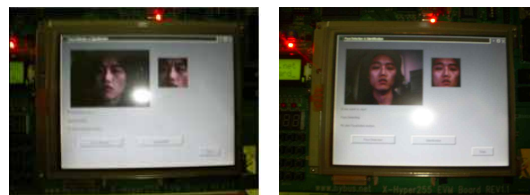
이 경우 얼굴 검출 데이터베이스를 구축할 때 얼굴을 숙이거나 고개를 좌우로 돌리는 데이터베이스가 없으므로 검출하지 못한다. 측면 얼굴이 아닌 고개를 좌우로 돌리는 경우에는 입력받은 이미지를 좌우로 약 5도씩 회전시켜 Haar-like feature를 사용하면 해결된다.

<그림 7>은 정지 영상이 아닌 실시간 영상에서 즉 실시간 임베디드 시스템에 맞는 동영상의 얼굴 검출을 보여 준다.



<그림 7> 실시간 얼굴 검출

카메라의 성능은 테스트 보드에서는 1초에 5-6프레임 정도이다. 또한 전체 320×240의 픽셀에 대해 Haar-like feature 알고리즘을 적용하게 되면 연산 속도가 느려 실시간 처리에 문제점이 있다. <그림 8>은 실시간 임베디드 리눅스 시스템에서 얼굴 검출 화면이다.

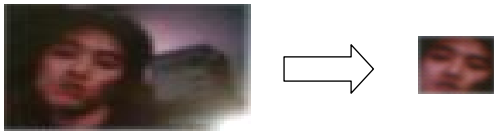


<그림 8> 임베디드 리눅스 시스템에서 얼굴 검출

임베디드 리눅스 시스템에서 웹 카메라와 얼굴까지의 거리에 따른 검출률에 대한 실험 결과(10cm~1m) 40cm~90cm까지 검출이 되었다.

PCA알고리즘만을 가지고 얼굴 인식을 실시한 경우에는 320×240에 대해서 연산을 실시하지만, 웹 카메라로부터 획득되어진 입력 이미지가 칼라이고 1채널당 256까지 인 8bit를 가지게 되므로 실제로는 320×240×3×8의 연산을 한다.

이런 이유로 웹 카메라의 입력 이미지로부터 배경을 제거하여 실시간 임베디드 리눅스 시스템을 구현하기 위하여 Haar-like feature를 이용하여 얼굴 검출을 실시하였으며 PCA의 입력 이미지는 <그림 9>와 같다.



<그림 9> Haar-like feature를 이용한 PCA 입력이미지

또한 배경 제거의 큰 장점은 인식률의 향상에 있다. 인식률 향상은 아래의 <표 2>에서 볼 수 있다.

<표 2> 데스크탑과 보드에서의 배경 유무에 따른 인식률

		인식 이미지 수	인식된 이미지 수	인식률(%)
데스크탑	배경을 포함하고 있는 이미지	250	132	52.8%
	배경을 제거한 이미지	250	214	85.6%
보드	배경을 포함하고 있는 이미지	30	14	46.6%
	배경을 제거한 이미지	30	19	63.3%

<그림 10>은 한사람에 3장씩 데이터베이스에 대한 그림이며 총 데이터베이스 중 일부만 표현한 것이다.

임베디드 리눅스 시스템에서 트레이닝을 하여 데이터베이스 구축을 할 시 공분산 행렬의 계산으로 인한 연산량이 많아져 시간이 오래 걸린다. <표 3>은 데이터 베이스 구축시 이미지 수에 따른 시간을 나타낸다.



<그림 10> 트레이닝 데이터베이스

<표 3> 트레이닝 이미지 수에 따른 연산 시간

이미지 수	10	20	30	40	50
연산 시간	20초	1분	2분 30초	4분	8분

얼굴 검출 시 거리에 따른 입력 이미지에 대해서 resize를 하여 70×80사이즈로 만드는 데 이때의 오차율로 인한 인식률을 <표 4>에서 알아본다. 인식할 이미지는 총 30명으로 하였다.

<표 4> 거리에 따른 인식률

이미지 수	40cm	50cm	60cm	70cm	80cm	90cm
연산 시간	53%	57%	58.9%	60%	66.5%	62%

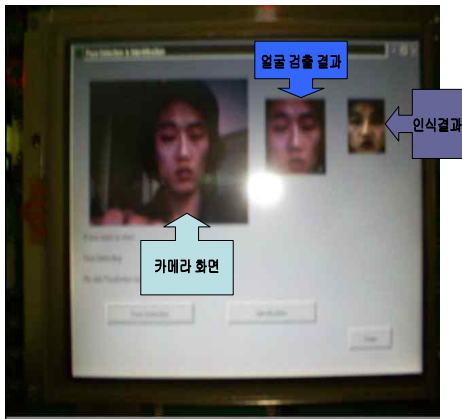
이런 결과를 초래하는 이유는 카메라의 성능 또한 좌우를 하겠지만, PCA입력 이미지가 작아 특징점을 분석을 할 때 트레이닝 데이터베이스의 유클리언 디스턴스가 비슷하기 때문이다 또한 80cm에서 인식률이 제일 높게 나온 결과의 이유는 데이터베이스 이미지 사이즈 70×80이 80cm에서 제일 유사하기 때문이다.

<표 5>에서 얼굴 인식 총 소요 시간에 대한 것을 보여 준다. 표를 보면 알 수 있듯이 총 인식하는 시간은 0.8초에서 1초 미만이다. 이는 실시간으로 처리가 가능하다는 결과를 가져온다.

<표 5> 얼굴 인식 총 소요 시간

	검출	인식시간	총
소요시간	0.5초	0.3-0.5	0.8-1초 미만

<그림 11>은 임베디드 리눅스 시스템에서 Qt 라이브러리를 이용하여 디자인을 한 얼굴 인식 시스템의 구현 화면이다. 각각의 버튼을 클릭을 하여 트레이닝, 얼굴 검출, 얼굴 인식을 할 수 있게끔 구현을 하였으며 시스템의 현재 어떤 알고리즘을 진행 상태인지 메시지로 user에게 알려 준다.



<그림 11> 임베디드 리눅스 시스템에서 얼굴 인식 구현 화면

VI. 결론 및 토의

본 논문에서는 Haar-like feature와 PCA를 이용한 실시간 임베디드 시스템에 대해서 논하였다. Haar-like feature와 AdaBoost 알고리즘을 적용하여 실시간으로 얼굴 검출이 가능케 하였다. 얼굴 검출의 연산이 끝난 후에는 Haar like feature의 출력을 PCA의 입력으로 하여 PCA의 알고리즘을 적용하였으며 인식률과 인식 속도를 높이기 위하여 배경을 제거한 이미지를 사용하였다. 얼굴 검출이 안되는 경우에 대한 문제점과 그 해결책을 논했으며, 임베디드 시스템에서의 인식률 향상을 실험했다. 하지만 보드에서의 인식률이 낮고, 일반 PC에서도 90% 이상을 못 넘긴다. 특히 얼굴 인식의 응용이 보편으로 가게 되면 이는 심각한 문제이다[11]. 이런 문제를 해결하기 위하여

다른 알고리즘과의 결합이든가 전처리를 통하여 극복 해 나가야 할 것이며, 지문 인식처럼 인식 환경을 극대화해 단순히 하는 방법도 연구 해 나가야 할 것이다. 인식률 높이기 위한 연구는 2D에서 뿐만 아니라, 3D에서도 연구가 활발히 진행되고 있으며 이 또한 연구해 볼 가치가 있다.

참고문헌

- [1] <http://biometrics.or.kr/>
- [2] Qt Reference Documentation(Open Source Edition), <http://doc.trolltech.com/3.3/index.html>
- [3] www.hybus.co.kr
- [4] 박성훈, 이재호, 김희율, "Haar-like feature/LDA를 이용한 얼굴 인식," 2004년 제 16회 영상처리 및 이해에 관한 워크샵 발표 논문집.
- [5] Randy Crane. "A Simplified Approach to Image Processing," Prentice Hall PTR, 1997.
- [6] Rainer Lienhart and Hochen Maydt. An Extended Set of Haar-like Features for Rapid Object Detection. IEEE ICIP 2002, Vol. 1, Sep. 2002, pp. 900-903.
- [7] Alexander Kuranov, Rainer Lienhart, and Vadim Pisarevsky. An Empirical Analysis of Boosting Algorithms for Rapid Objects With an Extended Set of Haar-like Features. Intel Technical Report MRL-TR-july02-01, 2002.
- [8] Kicheon Hong, Jihong Min, Wonchan Lee, and Jungchul Kim. Real Time Face Detection and Recognition System Using Haar-Like Feature/HMM in Ubiquitous Network Environments. ICCSA 2005, LNCS 3480, 2005, pp. 1154-1161.
- [9] Raychaudhuri, S., Stuart, J. M. and Altman, R. B. Principal components analysis to summarize

microarray experiments: application to sporulation time series. Pacific Symposium on Biocomputing 2000.

[10] Lindsay I Smith. A Tutorial On Principle Component Analysis. February 26, 2002.

[11] 장혜경, "PCA 기반 LDA 혼합 알고리즘을 이용한 실시간 얼굴인식 시스템 구현," 대한전자공학회, 2005.



홍기천
Hong, Ki Cheon

1998년~현재
수원대학교 정보통신공학과 교수
1994년 스티븐스 공과대학 전자공학과(EE) 공학박사
1988년 스티븐스 공과대학 전자공학과(EE) 공학석사
1985년 성균관대학교 전자공학과 (공학사)
1994년~1998년
삼성 반도체 기흥연구소 및 산호세 연구소 책임연구원
1992년~1993년
벨통신연구소(Bellcore) 연구원
1988년~1992년
ATI(Advanced Telecommunications Institute) 연구 조교

관심분야 : MPEG기반 시스템, 임베디드 실시간 시스템, 영상 처리 및 압축, 멀티미디어 프로세서 개발, 패턴 인식, 컴퓨터인터페이스
E-mail : kchong@suwon.ac.kr

논문접수일 : 2010년 4월 28일
수정일 : 2010년 5월 10일(1차), 5월 20일(2차)
게재확정일 : 2010년 5월 25일

■ 저자소개 ■



김정철
Kim, Jung Chul

2007년~현재
하이트루스시스템즈 주임 연구원
2007년 수원대학교 일반대학원 정보통신공학과 석사 졸업
2005년 수원대학교 정보통신공학과(공학사)

관심분야 : 멀티미디어시스템, 영상인식, 영상처리, 네트워크
E-mail : kjc3342@naver.com



허범근
Heo, Bum Geun

2010년~현재
수원대학교 일반대학원 정보통신공학과 석사과정
2010년 수원대학교 정보통신공학과(공학사)

관심분야 : 영상 처리, 객체 추적
E-mail : heo0928@suwon.ac.kr



신나라
Shin, Na Ra

2010년~현재
수원대학교 일반대학원 정보통신공학과 석사과정
2010년 수원대학교 정보통신공학과(공학사)

관심분야 : 영상 처리, 패턴인식
E-mail : nara9n@suwon.ac.kr