

SVM-SMO와 Pan-Tilt 웹카메라를 이용한 실시간

얼굴 추적과 얼굴 인식

이호근^o, 김영훈, 이지근, 정성태
원광대학교 컴퓨터공학과

{hghome^o, bbong09, lcg74, stjung}@wonkwang.ac.kr

RealTime Face Tracking and Recognition using SVM-SMO with a Pan-Tilt Web-Camera

Hogeun Lee^o, Myonghun Kim, Chigeun Lee, Sungtae Jung
Dept. of Computer Engineering, Wonkwang University

요 약

웹카메라로부터 입력된 비디오 영상으로부터 실시간 얼굴 인식은 빠르고 정확한 시스템이 요구된다. 따라서 본 논문에서는 객체 분류 기법인 SVM을 이용하여 실시간 다중 얼굴 인식이 가능한 시스템 구현에 중점을 두었다. 본 논문은 얼굴 skin/non-skin 정보를 이용한 얼굴 후보 영역의 검출 단계, 얼굴/비얼굴의 검출 단계, 그리고 얼굴의 인식 단계로 구성되어 있다. 각각의 단계별 SVM을 적용하였고 각 SVM은 오프라인상의 학습 부분과 온라인상의 테스트 부분으로 구성되어 있고, SVM의 QP 최적화 문제를 해결하기 위해 학습 알고리즘인 SMO를 적용하였다. 팬(Pan)-틸트(Tilt) 제어가 가능한 저가형 웹카메라를 이용하여 자동으로 얼굴 위치를 추적, 이동하면서 얼굴 인식을 수행하였다.

1. 서 론

최근 들어 컴퓨터 비전은 다양한 분야에서 활발하게 연구가 진행되어 왔다. 특히, 얼굴 인식은 신분 증명, 보안 시스템, 인사 자료, 회원 관리, 비디오 검색과 같은 실생활에 이용하기 위한 연구가 두드러지게 나타나고 있다. 인간의 얼굴에 대한 정보는 매우 다양하게 이용 될 수 있어, 특정 사람의 얼굴을 인식하는 적용에서부터 표정, 감정 인식 등에 활용 영역이 방대하다. 얼굴은 얼굴의 움직임, 크기 변화, 얼굴 표정, 헤어스타일, 타 객체와의 겹침, 조명, 카메라의 기계적인 특성 등에 따라 다양하게 나타날 수 있다. 그로인해, 얼굴 인식 일련의 과정들은 다양한 방법에 의해 연구되고 있다. 얼굴 영역의 검출 기법에는 피부색/밝기 기반 방법, 고유 함수 방법, 신경망을 이용한 방법, 특징 기반 방법, 혼합 기법 등이 있다. 얼굴 인식 기법은 PCA(Principal Component Analysis), LDA(Linear Discriminant Analysis), ICA (Independent Component Analysis), Neural Network, LFA(Local Feature Analysis), Wavelet, SVM(Support Vector Machine) 등이 있다[1][2][5]. 본 논문에서는 웹카메라의 영상으로부터 다양한 크기의 얼굴 검출을 위한 전체 이미지의 스케일 회피하고, 빠르고 정확한 얼굴 영역을 찾기 위해서 얼굴 피부색 정보와 SVM을 이용하여 실시간 얼굴 검출과 추적, 그리고 얼굴 인식을 수행하였다.

2. 소 개

2.1 시스템 구성

본 논문의 시스템은 웹카메라의 동영상 내에서 얼굴 후보 영역의 위치와 크기를 획득하고, 획득된 영역에 대하여 얼굴인지 아닌지를 검증한 다음에, 얼굴의 고유 특징을 가지고 분류 및 인식을 수행한다. 시스템 구성은 그림 1과 같이 3단계의 과정으로 이루어져 있다. 1단계에서는 모션 정보와 조명의 영향을 줄이기 위한 전처리 기법을 이용하였고, 속도가 빠른 선형 SVM을 이용하여 skin 픽셀과 non-skin 픽셀을 구별하기 위해 학습과 검출을 수행하였고, 검출된 skin 픽셀에 대한 레이블링을 통해 얼굴 후보 영역을 검출하였다. 2단계에서는 검출된 얼굴 후보 영역에 대해 이미지 크기를 정규화하여 특징 벡터를 만들어 비선형 SVM의 학습과 검출을 수행하였고, 팬(Pan)-틸트(Tilt) 기능을 가진 웹카메라를 이용하여 얼굴을 자동 추적하였다. 3단계에서는 얼굴로 검증된 영역에 대해 멀티 클래스 비선형 SVM을 사용하여 얼굴의 학습과 인식을 수행하였다.

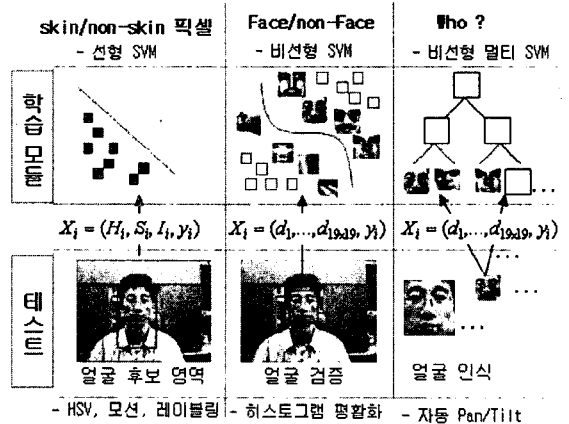


그림 1 시스템 구성도

2.2 SVM(Support Vector Machine)

SVM은 1995 Vapnik[3]에 의해 이진패턴 분류 문제를 해결하기 위해 제안된 학습방법으로 각 범주의 대표적인 특징을 갖는 벡터를 이용하여 최적의 하이퍼분리면(Hyperplane)을 찾아서 두 범주를 분류하는 기법이다. SVM은 인식 성능과 처리 속도 성능이 매우 뛰어나 최근 에 얼굴인식, 문자인식, 문서 분류 등 다양한 분야에 적용되고 있다[3][4][5].

그림 2(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 2(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 최적으로 한다.

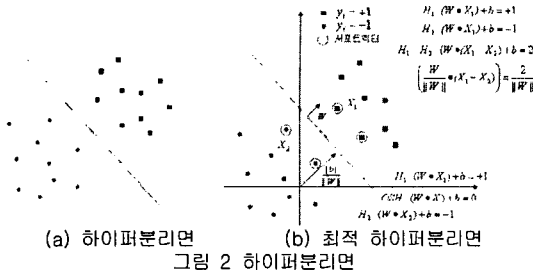


그림 2(b)에서 하이퍼플레인들을 벡터의 내적으로 표현하면 식 1과 같고 W 는 하이퍼플레인과의 직교하는 가중치 벡터(Weight Vector)이고, b 는 기준값(Threshold)이다.

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

$$f(X) = \sin((W \cdot X_i) + b) \quad \text{식 2}$$

2.2.1 선형 SVM

그림 2(b)와 같이 선형 분리가 가능한 경우에는 식 3과 같은 최적화 문제가 된다.

$$\max \frac{2}{\|W\|} \Rightarrow \min \frac{\|W\|}{2}$$

subject to $y_i(X_i \cdot W + b) - 1 \geq 0 \forall i$ 식 3

최적화 문제에서 함수의 최소 또는 최대 문제를 라그랑지(Lagrange) 함수로 유도하면 식 4와 같은 원문제(Primal)가 된다.

$$\min L_P = \frac{1}{2} \|W\|^2 - \sum_{i=1}^N \alpha_i y_i (W \cdot X_i + b - 1)$$

subject to $\alpha_i \geq 0, i=1, \dots, N$ 식 4

식 4에 대해서 울프 쌍대문제(Wolfe Dual Problem)로 유도하고 KKT(Karush-Kuhn-Tucker)조건을 적용하면, 식 5를 제약조건으로 하고 식 4를 최대화하는 최적화 문제가 된다[4][5].

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

subject to $\sum_{i=1}^N \alpha_i y_i = 0, 0 \leq \alpha_i \leq C$ 식 6

식 5,6에서 라그랑지 계수 α_i 는 2차 프로그래밍 문제(Quadratic Programming:QP)의 해[6]이며, $W = \sum_{i=1}^N \alpha_i y_i X_i$ 이고, C 는 마진 내의 객체를 허용하기 위한 페널티 값이다. $0 < \alpha_i < C$ 이면 SV가 되고, $\alpha_i = C$ 이면 마진에 위치하고, $\alpha_i = 0$ 이면 그밖에 위치한다.

2.2.2 비선형 SVM

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.

$$f(X_i) = \phi(X_i) \cdot W + b \quad \text{식 7}$$

식 7은 하이퍼플레인의 일반식이고 $\phi(X)$ 는 X 의 기저함수로써 보다 높은 차원의 맵핑을 수행하는 함수이다. 식 8과 같이 두

벡터의 내적을 나타내기 위한 커널 함수 $K(X, X_i)$ 를 이용하여 투영시켜서 선형 분류문제로 해결한다.

$$K(X, X_i) = \phi(X) \cdot \phi(X_i) \quad \text{식 8}$$

맵핑함수를 사용한 라그랑지 듀얼 문제는 식 9와 같다.

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

subject to $\sum_{i=1}^N \alpha_i y_i = 0, 0 \leq \alpha_i \leq C$ 식 9

일반적으로 사용되는 커널 함수는 식 10과 같다.

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

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

Sigmoidal : $K(X, X_i) = \tanh(kX \cdot X_i - \delta)$ 식 10

3. 제안된 얼굴 검출과 인식

웹카메라로부터 획득된 영상은 이전 프레임과 현재 프레임의 차이가 적으므로 영상내의 모션 정보를 이용하여 객체 움직임이 있을 때, 해당 프레임에서 얼굴 후보 영역 검출, 얼굴 검출, 얼굴 인식과 자동 추적 과정을 수행하였다.

3.1 얼굴 후보 영역 검출

본 논문에서는 얼굴 피부색을 이용하여 얼굴 후보 영역을 검출하였다. 속도가 빠른 선형 SVM을 이용한 skin/non-skin 픽셀에 대한 학습과 분류, skin으로 분류된 픽셀에 대한 영역 레이블링 과정을 수행한다.

3.1.1 skin/non-skin 픽셀 학습과 분류

이미지 내에서 skin 픽셀을 찾기 위한 SVM의 데이터는 식 11과 같이 얼굴 피부색에 효과적인 HSI 컬러모형으로 변환하였고 조명에 영향을 덜 받기 위해서 영감 I 를 정규화하였다.

$$I = \frac{1}{3}(R+G+B), \quad \text{scale} \times \frac{I_{\text{original}} - I_{\text{min}}}{I_{\text{max}} - I_{\text{min}}}$$

$$S = 1 - \frac{3}{R+G+B} \min(R, G, B)$$

$$H = \cos^{-1} \left\{ \frac{\frac{1}{2} [(R-G) + (R-B)]}{\sqrt{[(R-G)^2 + (R-B)(G-B)]}} \right\} \quad \text{식 11}$$

$$X_i = (H_i, S_i, I_i, y_i), \quad y_i \in \{\text{skin}, \text{non-skin}\} \quad \text{식 12}$$

SVM에 적용되는 데이터 집합은 식 12와 같이 구성하였고, skin/non-skin 테스트 과정은 한 프레임내의 각각의 모든 픽셀에 대해서 분류를 수행하였다.

3.1.2 얼굴 후보 영역 레이블링

skin 픽셀을 검출한 다음에는 얼굴 후보 영역의 크기와 위치를 찾기 위해 skin 픽셀에 대해 영역 레이블링을 수행하였다. skin 픽셀과 non-skin 픽셀 사이에서 연관성을 낮추고 잡음을 제거하는 침식 연산과 객체의 최외각 픽셀을 확장하는 팽창 연산을 수행하였고, 레이블링 알고리즘으로는 주변에 이어져 있는 모든 픽셀에 대해 루프를 반복하는 재귀 알고리즘과 동치 테이블을 이용하는 순차적 알고리즘이 있다. 재귀 알고리즘은 U자형 구조에서 루프를 많이 해야 하기 때문에 빠른 계산 시간을 위해 순차적 알고리즘을 적용하였다.



(a) 입력영상 (b) skin 픽셀 (c) 레이블링
그림 3 얼굴 후보 영역 레이블링

레이블링된 skin 픽셀 영역은 사람의 얼굴과 목 부분을 포함되어 있기 때문에 가로, 세로의 비율 ($\sqrt{5}-1$)로 가정하여 목 영역을 제거하였다. 얼굴 후보 영역 내의 이미지에 대해서 히스토그램

평화를 적용 후 크기를 19×19로 정규화하였다.

3.2 얼굴 검출

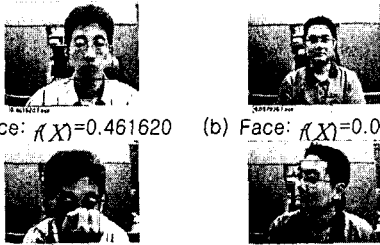
얼굴 후보 영역이 얼굴 인지 아닌지를 판단하기 위해서 비선형 SVM 분류기를 이용하였고, 2차 다항식 커널과 페널티 C=200으로 설정하였다. 학습과 테스트 단계에서 얼굴/비얼굴 이미지를 19×19크기로 정규화하여 수행하였다.

$X_i = (d_1, \dots, d_{19 \times 19}, y)$, $y_i \in \{face, non-face\}$ 식 14
 얼굴/비얼굴 학습 데이터는 MIT CBCL 데이터베이스를 사용하였다[8]. CBCL은 학습 데이터로 2,429개의 얼굴과 4,548개의 비얼굴로 구성되어 있고, 테스트 데이터로 472개의 얼굴과 23,573개의 비얼굴의 이미지로 구성되어 있다.



(a) 얼굴(19×19) (b) 비얼굴(19×19)
 그림 4 얼굴 검출을 위한 학습 데이터 샘플

얼굴 검출 테스트 단계에서는 검출된 얼굴 후보 영역들을 가장 큰 영역부터 차례로 얼굴/비얼굴 검증을 수행한다. 그림 5는 얼굴 검출 예제로 테스트 출력 값이 양수이면 얼굴로 분류된다.



(c) non-Face: $K(X) = -9.70859$ (d) non-Face: $K(X) = -0.596883$
 그림 5 얼굴 후보 영역 검증 예제

3.3 얼굴 인식

검출된 얼굴 영역은 얼굴 인식을 위해 2차 다항식 커널과 페널티 C=200이 적용된 비선형 SVM를 사용하여 학습과 인식을 수행하였다. SVM은 이진 분류기로서 n개의 클래스를 분류하기 위한 방법으로 일대다방법(one-vs-all approach)과 쌍방향법(Pairwise approach)이 있다. 일대다방법은 클래스 수만큼의 이진 SVM을 이용하여 각 이진 분류기는 한 클래스와 나머지 다른 클래스들을 학습하는 방법이다. 각각의 이진 SVM의 결과로 가장 높은 값을 갖는 경우 해당 클래스로 분류하게 된다. 쌍방향법은 모든 클래스에 대해 가능한 모든 쌍을 구성하여 각각에 대해서 이진 SVM을 만든다. 하나의 이진 SVM은 오직 두 클래스에 관련된 샘플에 대해서 학습을 하게 된다. SVM 분류기는 이진트리 형태로 구성되어 각각 두 개의 SVM의 경쟁(voting)을 통해 트리의 최상의 클래스가 선택된다. n-class에서 필요한 SVM의 수는 $n(n-1)/2$ 개가 된다. 본 논문에서는 학습 샘플이 적은 경우에 효과적이고, 한 번에 모든 클래스를 학습하는 일대다방법으로 구현하였다.



(a) 얼굴 인식 학습 샘플



(b) 얼굴 인식 예제
 그림 7 실시간 얼굴 인식

3. 얼굴 자동 추적에 위한 카메라 팬(Pan)/틸트(Tilt)

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 $\pm 90^\circ$ ($90^\circ/\text{sec}$), 틸트 $\pm 30^\circ$ ($90^\circ/\text{sec}$)의 사양을 보유하고 있다. 스마일캠의 이동 제어를 하기 위해 시리얼 포트(COM)를 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.

4. 실험 및 결론

본 논문의 실험은 사용자의 다중 얼굴을 자동 추적하고 인식하기 위한 실시간 시스템 구현에 중점을 두었다. 실시간 처리에 효율적인 방법들을 적용하였고 인식률을 높이기 위해 분류 성능이 뛰어난 SVM를 사용하였다. MS Windows XP 환경, Pentium 4 3.0GHz, MS Visual C++ 6.0을 사용하였다.

SVM은 John C. Platt의 SMO(Sequential Minimal Optimization) Pseudocode 프로그램을 이용하여 구현하였다 [6]. QP 문제를 작은 QP 문제로 분해해서 분석(Analytic)적으로 해결하는 SMO는 아주 적은 코드로 구현이 가능하고, 전체적으로 빠른 수렴 성능을 보였다. 또한 웹카메라로부터 이미지 획득을 위해 FrameGrabber 클래스, 팬-틸트 제어를 위해 통신 COMThreadComm 클래스, 그리고 이미지 처리 및 뷰를 위한 ImageProcess 클래스를 만들어 사용하였다.

표 1. 실험 결과

구분	인식률	평균 수행시간
skin 픽셀 분류	96.7%	0.031
얼굴 검증	97.2%	0.023
얼굴 인식	98.5%	0.124

SVM의 학습 과정에서 위해 사용자가 직접 학습 샘플데이터, 매개변수(Parameter) 값, 커널 함수를 미리 결정해야 한다. 그러므로 최적의 성능을 구현하기 위해 다양한 데이터를 SVM에 이용하여 실험에 의한 최적의 값을 선택하였다. 본 논문에서 제안된 skin 정보를 이용한 SVM의 적용은 기존의 입력 이미지의 크기를 줄여가며 스캔하는 방법보다 계산 비용을 줄였기 때문에 실시간 얼굴 검출이 향상되었고, 팬-틸트 웹카메라를 이용하여 움직이는 얼굴을 추적하므로 얼굴 인식의 성능을 향상시켰다.

5. 참고

[1] W. Zhao, R. Chellappa, P.J. Phillips, A. Rosenfeld, "Face Recognition: A Literature Survey", ACM Computing Surveys(CSUR), v.35 n.4, pp.399-458, 2003.
 [2] Ming-Hsuan Yang, David Kriegman, and Narendra Ahuja, "Detecting Faces in Images: A Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 24, no. 1, pp. 34-58, 2002
 [3] V. Vapnik, "The Nature of Statistical Learning Theory", Springer-verlag, New York, 1995.
 [4] C.J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", in Data Mining and Knowledge Discovery, v.2 n.2, pp.121-167, 1998.
 [5] E. Osuna, R. Freund, F. Girosi, "Training Support Vector Machines: An application to face detection", Proceeding IEEE. CVPR, pp.130-136, 1997
 [6] Platt, J.C., "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines", Microsoft Research Technical Report MSR-TR-98-14, 1998.
 [7] <http://research.microsoft.com/users/jplatt/smo.html>
 [8] <http://cbcl.mit.edu/cbcl/software-datasets>