

Pan-Tilt 카메라를 이용한 실시간 얼굴 검출 및 추적 시스템

임옥현^o 김진철 이배호
전남대학교 컴퓨터정보통신공학과
{ohlim^o, jckim, bhlee}@chonnam.ac.kr

The Real-Time Face Detection and Tracking System using Pan-Tilt Camera

Okhyun Lim^o Jinchul Kim Baeho Lee
Dept. of Computer Engineering, Chonnam National University

요 약

본 논문에서는 웨이블릿을 이용한 알고리즘으로 얼굴을 검출하고 검출된 얼굴을 움직이는 Pan-Tilt 카메라상에서 추적하는 방법을 제안하고자 한다. 우리는 얼굴 검출을 위해 다섯 종류의 간단한 웨이블릿을 사용하여 특징을 추출하였고 AdaBoost(Adaptive Boosting) 알고리즘을 이용한 계층적 분류기를 통하여 추출된 특징들 중에서 얼굴을 검출하는데 강한 특징들만을 모았다. 이렇게 만들어진 특징집합들을 이용하여 입력받은 영상에서 초당 20프레임의 실시간으로 얼굴을 검출하였고 영상에서 얼굴 위치와 Pan-Tilt 카메라 위치를 계산하여 실시간으로 움직임을 추적하는데 성공하였다.

1. 서 론

얼굴 검출 및 추적 시스템은 영상감시 시스템, 원격회의 시스템, 얼굴 인식 시스템 등 다양한 분야에 응용될 수 있는 기술이다. 그러나 여전히 복잡한 주위 환경, 다양한 조명 조건, 여러 가지 자세 문제로 인하여 실시간 얼굴 검출에 많은 어려움이 있다.

기존의 신경망을 이용한 방법, 특징 기반 방법, 색상기반의 방법들이 많이 연구되고 있으나 신경망을 이용한 방법은 데이터 훈련에 대한 많은 시간이 소요되는 단점이 있고, 특징 기반 방법은 얼굴 특징들을 정확하게 계산하는데 많은 시간이 소요되는 단점이 있으며, 색상 기반은 연산량은 적으나 주위 환경과 조명 변화에 민감한 약점을 가지고 있다. 특히 실시간으로 얼굴을 검출하고 추적하기 위해서는 빠른 연산 속도로 정확한 얼굴을 검출을 할 수 있는 방법이 필요하다. 최근에 간단한 웨이블릿을 기반으로 한 얼굴 검출은 데이터 훈련에 대한 시간을 신경망보다 적게 걸리면서도 실제 검출에 있어서는 주위 환경과 조명변화에 강인하다는 장점이 있어서 실시간 얼굴 검출에 많이 활용되고 있다[1][2].

본 논문에서는 다섯 종류의 간단한 웨이블릿을 통하여 얼굴에서 특징을 추출하고 AdaBoost(Adaptive Boosting) 알고리즘을 이용한 계층적 분류기를 통하여 추출된 특징들 중에서 얼굴을 검출하는데 강한 특징들만을 모아 얼굴을 검출하는 알고리즘을 제안한다. 또한 광범위에 시야를 Pan-Tilt 카메라를 이용하여 동적으로 확보하고 이것을 통하여 입력받은 영상에서 실시간으로 얼굴을 추적하는 방법을 제안한다.

본 논문의 구성은 2장에서 얼굴 검출 알고리즘 설명, 3장에서 얼굴 추적 알고리즘 설명, 4장에서 제안한 알고리즘을 이용한 실험결과, 마지막으로 5장에서는 결론으로 이루어져있다.

2. 얼굴 검출 알고리즘

얼굴 검출 알고리즘은 웨이블릿을 이용한 얼굴 특징의 추출과 AdaBoost 알고리즘을 이용한 얼굴 특징의 분류로 이루어진다.

2.1 얼굴 특징의 추출

본 논문에서는 각각의 얼굴에 특징을 추출하기 위해 다섯 종류의 웨이블릿을 이용하였다. 웨이블릿 기반의 특징 계산은 픽셀 기반의 특징 계산보다 간단하고 효율적이다.

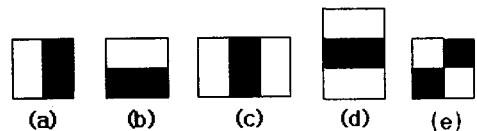


그림 1. 다섯 종류의 웨이블릿

제안된 각각의 웨이블릿을 20*20크기로 구성된 훈련 영상에 적용하여 모든 가능성을 갖는 초기의 특징 집합을 이루게 된다. 초기의 특징 집합은 픽셀 기반에서 픽셀과 같은 기본 연산 파라미터가 된다.

웨이블릿에 의해 생성된 특징들은 특징값으로 변환되어 분류기를 통한 학습에 이용된다. 그러나 특징의 수가 많기 때문에 많은 연산 시간이 필요하게 된다. 본 논문에서는 SAT(Summed-Area Tables) 방법을 사용하여 특징값을 빠르게 계산하였다. SAT는 영상처리에서 사용되는 영역의 합 테이블과 유사한 형태이고 중복 연산을 최소화하여 연산 속도를 빠르게 한다[3].

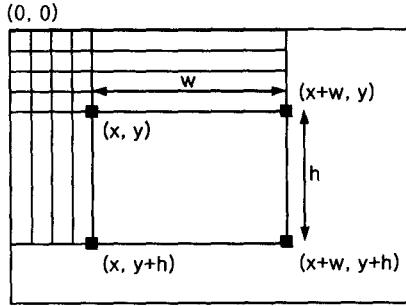


그림 2. SAT(Summed-Area Tables)

SAT는 영상 내에 존재하는 특정한 좌표 영역(x, y)의 값은 입력 영상의 원점(0, 0)에서부터 좌표 영역(x, y)를 포함하는 사각형 영역 내에 존재하는 모든 픽셀 값을 더한 값이며, 식(1)과 같이 정의된다.

$$S(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y') \quad (1)$$

하나의 사각형 특징값(RecS)은 식(2)와 같이 정의된다. 이 값은 미리 계산된 SAT테이블을 이용하여, 네 개의 SAT를 사용하여 계산된다.

$$RecS(x, y, w, h) = S(x + w, y + h) - S(x + w, y) - S(x, y + h) + S(x, y) \quad (2)$$

선택되어진 전체 영역의 특징값(RecF)은 식(2)를 사용하여 사각형 픽셀값(RecSw)의 평균과 검정색 사각형 픽셀값(RecSb)의 평균에 대한 차로 구하며, 식(3)과 같이 정의된다.

$$RecF(x, y, w, h) = \frac{1}{m} \sum_m RecSw - \frac{1}{n} \sum_n RecSb \quad (3)$$

2.2 얼굴 특징의 분류

본 논문에서 AdaBoost(Adaptive Boosting)알고리즘을 이용한 계층적 분류기로 얼굴 특징을 분류한다. AdaBoost 알고리즘의 기본 개념은 약한 분류기(Weak classifier)를 선형적으로 결합하여 최종적으로 높은 검출 성능을 가진 강한 분류기(Strong classifier)를 생성하는 것이다. 이는 분류기의 가중치를 바꾸어가면서 단계적으로 반복하여 적용하고 각 단계마다 특징 집합 중에서 얼굴을 검출하는데 결정적인 역할을 하는 특징만을 남기고 나머지는 제거하는 방식이다[4].

분류기는 얼굴과 비얼굴 패턴으로 구성된 훈련 영상 집합과 특징집합을 이용하여 학습된다. 입력벡터 X는 특징 집합이며, x_i 는 하나의 웨이블릿 특징 마스크이다. 출력 Y는 +1, -1 값을 가지며, 각각 얼굴과 비얼굴을 나타낸다.

약한 분류기 h_t 가 하나의 웨이블릿 특징 마스크 x_i 를 이용하여 훈련패턴을 분류한다. 이때 잘못 분류된 훈련패턴은

가중치 $D_t(i)$ 를 증가시키고, 옳게 분류된 훈련패턴은 $D_t(i)$ 를 감소시키는 과정을 반복하면서 최소의 에러율을 나타낼 때의 웨이블릿 특징만을 선택한다.

약한 분류기는 다시 처음부터 입력 데이터를 받아들이고 이런 과정을 반복하면서 단계별로 점점 얼굴과 비얼굴을 정확하게 구별할 수 있는 특징 집합들만을 남기고 이런 과정을 통하여 강한 분류기를 생성한다. 강한 분류기는 약한 분류기의 선형적인 결합 형태로 이루어졌다.

Given: $(x_1, y_1), \dots, (x_m, y_m)$
 where $x_i \in X, y_i \in Y = \{-1, +1\}$

Initialize $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

- Train weak learner using distribution D_t .
- Get weak hypothesis $h_t: X \rightarrow \{-1, +1\}$ with error.
- Choose $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$.
- Update:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

$$= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is a normalization factor
(chosen so that will be a distribution).

Output the final hypothesis:

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$$

그림 3. AdaBoost 알고리즘

3. 얼굴 추적 알고리즘

본 논문에서는 20*20크기의 영상으로 분류기를 학습하였으나 다양한 크기의 얼굴 영상을 검출하기 위해 다중해상도 피라미드를 사용하였다.

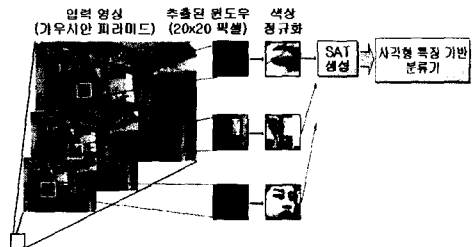


그림 4. 카메라 입력 영상의 전처리

움직이는 카메라를 이용하여 얼굴을 추적할 때는 프레임 간의 얼굴의 움직임은 방향, 흐름정보 등을 이용한다. 그러나 카메라와 추적할 얼굴이 같이 움직인다면 알고리즘이 복잡해지고 이로 인하여 초당 20프레임 이상을 지원해야하는 실시간 추적은 어렵게 된다. 따라서 본 논문

에서는 프레임간의 얼굴의 크기와 얼굴의 위치정보, Pan-Tilt 카메라의 위치정보 등 최소한의 정보만을 이용하여 실시간으로 얼굴을 추적하였다.

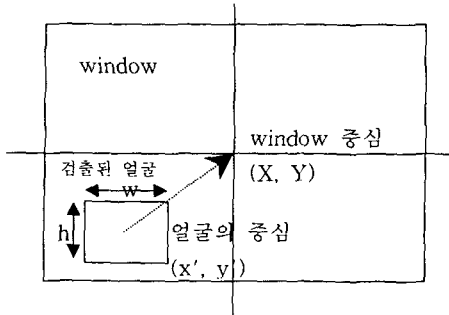


그림 5. 영상에서 얼굴 추적

실시간 얼굴 추적은 검출된 영역의 중심을 전체 영상의 중심으로 옮기는 방법을 기본 개념으로 사용한다. 영상에서 검출된 얼굴의 위치정보와 크기 정보를 이용한 얼굴의 중심 계산은 식(4)와 같이 나타낼 수 있다.

$$(x', y') = \left(\left(\frac{x+w}{2} \right) * scale, \left(\frac{y+h}{2} \right) * scale \right) \quad (4)$$

얼굴 추적은 얼굴의 중심을 영상의 중심에 위치하도록 하면서 이루어진다. 영상의 중심과 얼굴의 중심의 위치 정보를 이용하여 Pan-Tilt를 구동하기 위한 파라미터를 결정한다. 이때 얼굴의 움직임을 부드럽게 하기 위하여 가우시안 분포의 가중치를 이용한다. 얼굴 검출 실패로 인한 얼굴 추적이 순간적으로 실패하였을 경우 이것을 보완하기 위하여 10프레임 단위의 얼굴 위치와 크기 정보를 저장하여 보존하였다.

4. 실험결과

본 논문에서 제안한 다섯 종류의 웨이블릿을 이용하여 MIT의 CBCL 얼굴 훈련 영상에서 초기의 특징 집합 42,000개를 생성한다.

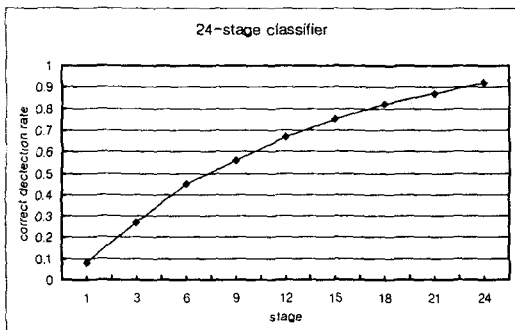


그림 6. 단계별 검출률

초기의 특징 집합 42,000개와 입력 받은 훈련 영상을 AdaBoost 알고리즘을 이용하여 분류기를 학습한다. 학습된 분류기는 그림6과 같이 24계층으로 이루어지고 초기의 특징 집합 42,000개 중에서 얼굴을 결정하는 특징 2,137개를 얻었다. 단계별로 검출률이 90% 이상이 될 때까지 분류기의 학습을 하였다.

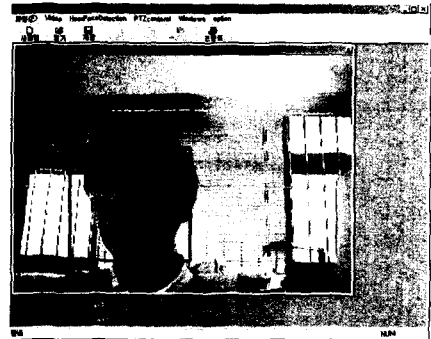


그림 7. Pan-Tilt카메라를 이용한 실험결과

Pan-Tilt를 이용한 광범위 시야에서 얼굴 검출 및 추적에서 초당 20프레임 이상의 속도와 80% 이상의 검출 및 추적 성공률을 보였다. 또한 복잡한 환경과 다양한 조명 조건에 관계없이 높은 검출률을 보였다.

5. 결론

본 논문은 다섯 종류의 웨이블릿을 이용하여 특징 집합을 생성하고 특징 집합 중에서 AdaBoost알고리즘을 이용한 분류기를 통하여 얼굴을 검출하는데 결정적인 영향을 미치는 특징 집합을 골라 단계적으로 학습하여 최종적으로 얼굴을 검출할 수 있는 알고리즘을 제안하였고 Pan-Tilt 카메라를 이용하여 광범위 시야에서 실시간으로 얼굴을 검출하고 추적하였다. 실험결과 초당 20프레임의 속도로 실시간 검출 및 추적이 가능하였고 복잡한 환경과 조명 조건하에서도 80% 이상의 성공률을 나타내었다.

참고문헌

- [1] Paul Viola, M. Jones, "Robust real-time object detection", *International Conference on Computer Vision*, 2001.
- [2] S.Z. Li, L. Zhu, Z.Q. Zhang, and H.J. Zhang, "Statistical Learning of Multi-View Face Detection", In *Proc. 7th European Conference on Computer Vision*, Copenhagen, Denmark. May 2002.
- [3] F. Crow, "Summed-area tables for texture mapping", In *Proceedings of SIGGRAPH*, Vol. 18(3), pp. 207-212, 1984.
- [4] Y. Freund, R.E. Schapire, "A Short Introduction to Boosting", *Journal of Japanese Society for Artificial Intelligence*, 14(5), pp. 771-780, 1999.