

얼굴 검출에 기반한 강인한 객체 추적 시스템

곽민석*

*한국디지털미디어고등학교

e-mail : kms7530@gmail.com

Face Detction Based Robust Object Tracking System

Min Seok Kwak*

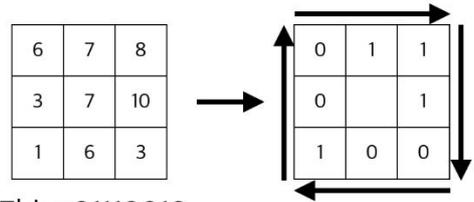
*Korea Digital Media High school

요 약

최근 컴퓨터 기술의 발전과 함께 임베디드 기기 또한 다양한 기능을 갖추기 시작했다. 본 연구에서는 최근 활발하게 진행되고 있는 영상센서를 사용한 임베디드 기기 등 자원이 적은 기기에서 효율적인 얼굴 추적 방식을 제안한다. 정확한 얼굴을 얻기 위하여 MB-LBP 특징을 사용한 얼굴 검출 방식을 사용했으며, 다음 영상에서 얼굴 객체 추적을 위하여 얼굴 검출시 얼굴 주변 영역(Region of Interest)을 지정하였다. 그리고 얼굴을 검출을 못하는 영상에서는 기존의 객체 추적 방식인 CAM-Shift 를 사용해 객체를 추적해 객체 정보의 손실 없이 정보를 유지 할 수 있도록 하였다. 본 연구는 기존 연구와의 비교를 통하여 객체 추적 시스템의 정확성과 빠른 성능을 확인하였다.

1. 서론

최근 사물인터넷의 유행으로 임베디드 기기 또한 각광을 받고 있다. 임베디드 기기의 발전과 함께 영상 센서를 사용한 영상 처리 분야 또한 주목을 받고 있다. 임베디드 기기를 사용한 영상 처리 분야는 자동차, 사물인터넷, 보안 등의 폭 넓은 분야에서 사용을 한다. 하지만 많은 자원을 요구하는 영상 처리 기술은 임베디드 기기에서 사용상 많은 제약이 따를 수 밖에 없다. 본 연구에서는 제한된 자원을 효율적으로 사용할 수 있는 방법을 제안한다. 전통적인 얼굴 검출 방법과 객체 추적 방법을 사용해 얼굴을 검출하여 추적 한다. Viola-Jones 가 제안한 Adaboost 를 사용한 얼굴 검출 방식은 전통적인 얼굴 검출 알고리즘으로 얼굴에서 학습되어 있는 정보를 사용해 정확한 얼굴 검출이 가능 하지만 다양한 각도의 얼굴 객체를 검출하기 힘들며, 기본적으로 기계 학습 알고리즘은 자원을 많이 사용해 객체 추적 방식에는 적합하지 않다.[2] 반면 전통적인 객체 추적 방식인 CAM-Shift 방식은 색상 정보를 기반으로 정확한 얼굴을 추적하기는 힘들지만 다양한 각도 변화에서 강인한 객체 추적을 할 수 있다.[4] 본 연구에서는 전통적인 얼굴 검출 기법과 객체 추적 기법을 사용해 각각의 알고리즘을 효율적으로 사용할 수 있는 객체 추적 시스템을 제안한다. 본 연구는 먼저 연구에서 사용한 MB-LBP 알고리즘과 CAM-Shift 알고리즘을 설명 한다.[4] 다음 장에서는 본 연구에서 제안하는 객체추적방식을 설명 한다. 그 후 기존 알고리즘과의 비교를 통한 성능 비교를 한 후 결론을 맺도록 한다.



2진수 : 01110010
10진수 : 114

(그림 1) LBP 작동 방식

2. MB-LBP (Multi-Scale Block LBP)

LBP(Local Binary Pattern)는 주로 영상에서는 효과를 주기 위해 만든 필터로 주로 사용된다. 최근 LBP 는 이용하여 얼굴 검출에도 널리 사용하며 얼굴 외에도 객체 검출에서도 강한 성능을 보여준다. 기존 LBP 는 3×3 인 블록에서 중심에 있는 픽셀을 기준으로 주변의 화소값이 크거나 같으면 1, 작으면 0 을 넣는다. 그 후 각 픽셀의 값들을 취합해 2 진수의 수를 10 진수로 나타낸다. 일반적인 (수식 1), (수식 2)와 같다. [13]

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(i_p - i_c)2^p$$

(수식 1) LBP 수식 1

$$s(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}$$

(수식 2) LBP 수식 2

이후 LBP 로 변환한 이미지의 히스토그램을 이용하여 얼굴인식을 하게 된다. 아래의 사진은 LBP 적용 후 히스토그램으로 표현한 것이다.



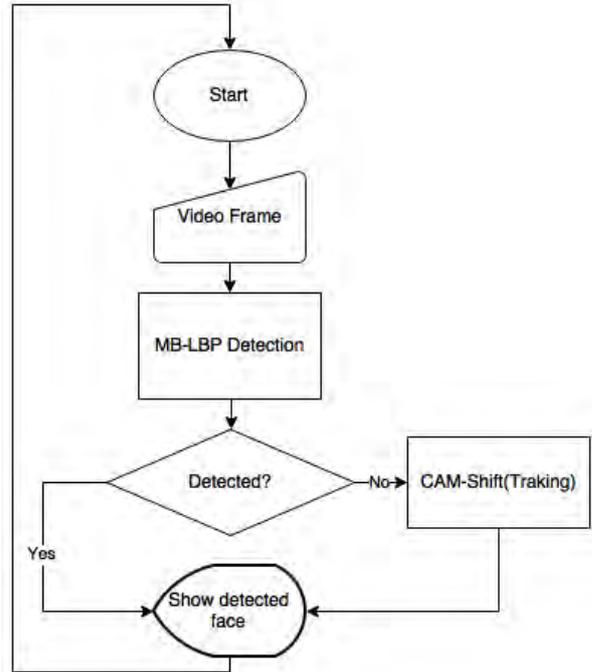
(그림 2) 얼굴인식에 대한 LBP 처리와 색상 정보

그러나 위의 사진 2 와 같이 기존의 LBP 를 적용한 히스토그램의 경우 픽셀 단위로 작업하기 때문에 국부적인 특징만 찾아낼 수 있다. 이 문제점을 해결 하기 위해 나온 것이 2007 년에 발표된 MB-LBP(Multi-Scale Block LBP)이다. MB-LBP 는 3×3 의 제한된 픽셀이 아닌 임의의 크기의 영상영역에서 3×3 의 블록으로 나누어 그 블록의 중심을 기준으로 계산하기 때문에 국부적으로 특징을 찾아내는 기존 LBP 와 비교해 큰 스케일의 텍스처 정보를 추출할 수 있다. 즉 픽셀 단위로 특징을 찾게 되는 LBP 를 확장하여 MB-LBP 는 픽셀이 아닌 블록단위의 연산을 하여 특징을 찾기 때문에 한 대상에 대해서 다양한 특징을 찾을 수 있게 된다.

3. CAM-Shift(Continuously Adaptive Mean-Shift)

CAM-Shift 는 크기의 유동성을 스스로 조절하는 기법을 사용하여 추적하는 객체의 크기가 변하며 색상 정보가 변할 경우 약한 모습을 보이는 하면 찾아내지 못하는 Mean-Shift 의 단점을 보완하여 나온 알고리즘이다. 기초적인 추적 방식은 색상 정보를 사용 하기 때문에 객체 추적을 위하여 먼저 지정 영역 주변의 추적할 객체의 관심영역을 가져야 한다. 여기서 고려하게 되는 요소는 초기 위치, 객체의 대칭 또는 비틀림, 회전, 원형 또는 사각형 그리고 크기를 고려한다. 그 후 관심 영역을 찾은 뒤 그 관심영역의 HSV 정보를 색상정보로 변환한다. 그리고 HSV 로 변환한 영상을 색상 정보로 변환해 객체 추적 정보로 사용 한다. 그리고 탐색 구간을 중심으로 객체를 탐색하는데 관심영역으로 지정된 곳과 비슷한 색상 정보를 찾게 된다. 그 후 색상 정보 변환을 한 영역에서 지정 영역의 색상 정보와 비교하여 물체를 추적한다. 이 추적 방식의 과정에서는 해당 영역에서 비슷한 영역을 가중치를 더 주게 되는데 추적하는 동안 이 작업을 계속하여 이동 벡터가 0 이 될 때까지 이동한다. 그 후 이동 벡터가 0 에 수렴하게 되는 경우 탐색을 멈추고 결과를 출력 한다.

4. 얼굴 검출에 기반한 객체 추적 시스템



(그림 3) 제안하는 추적기 전체 흐름도

처음의 영상이 들어오게 되면 그 영상에서 지정 영역을 사용하여 MB-LBP 얼굴 검출을 시도한다. 얼굴 검출에 성공하면 검출된 얼굴 정보를 추적 정보로 사용해 검출 영역을 지정한다. 여기에서 지정 영역을 사용 하는 이유는 한 영상에서 검색 시도 횟수를 줄이기 위해서 얼굴 검출이 될 가능성이 높은 부분에서만 검색하기 위해서다. 검출 영역 지정 작업을 통해서 객체 검출 시 영상의 모든 부분에서 검출을 시도 하는 것보다 높은 성능을 낼 수 있다. 그러나 만약 얼굴 검출을 시도했으나 검출되지 않으면 객체 추적을 시도한다. 추적을 시도할 때는 전 영상에서 저장된 얼굴 정보를 사용하여 추적하게 된다. 얼굴 추적의 경우 또한 찾아낸 얼굴 정보를 새로 갱신하여 다음 영상에서 객체를 찾는데 사용한다. 위의 그림 3 은 전체 흐름도를, 아래의 추적시스템에 대한 수식이 다.

$$f(k+1) = \begin{cases} \text{if } x > 0 & \sum_{p=0}^{P-1} s(i_p - i_c) 2^p \\ \text{if } x = 0 & \frac{\sum_{i=1}^{n_h} X_i \omega_i g \left(\left\| \frac{Y_0 - X_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n_h} \omega_i g \left(\left\| \frac{Y_0 - X_i}{h} \right\|^2 \right)} \end{cases}$$

(수식 3) 제안하는 추적기시스템

f(k)를 이전 영상이라고 할 때 f(k+1)을 얻기 위해서는 다음 (수식 3)과 같다. x보다 많은 경우 얼굴 검출 정보를 사용해 f(k+1)을 구하고 x의 값이 0 과

같은 때 객체 추적 방식을 사용해 $f(k+1)$ 을 구한다. 여기서 i 는 얼굴 추적 때 탐색하는 영상이다. 그리고 p 는 얼굴 검출 때 탐색하는 영상이다. p 는 전체 영상에서 찾는 검출 방법 보다 작다.

5. 실험 결과



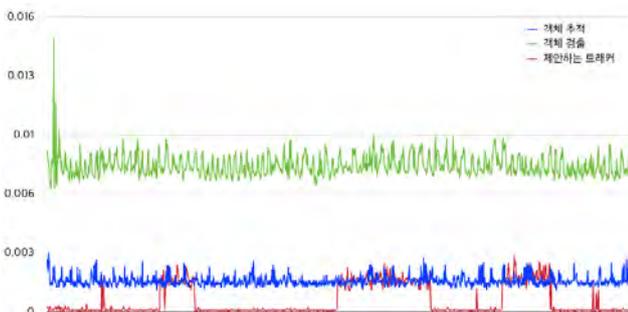
(그림 4) 기존 객체 검출과 제안하는 추적기



(그림 5) 기존 객체 추적과 제안하는 추적기

사진 4의 위의 행과 같이 영상에서 겹침이 발생한 경우 이번에는 객체를 찾지 못한다. 이 경우 객체 추적 도중 실패하게 되지만 그러나 본 연구에서 제시하는 방식은 사진 4와 같이 얼굴 검출이 불가능한 구간에서는 객체 추적을 하게 된다. 사진 5의 위의 행은 객체 추적이다. 객체 추적의 원래 문제점은 객체 검출보다 정확성이 떨어지는 단점이 있는데 반해, 제안하는 추적기는 객체 검출이 가능한 구간에서는 객체 검출을 시도할 함으로서 객체추적보다는 좀 더 정밀한 검색을 시도할 수 있게 된다. 다음 실험데이터는 본 연구에서 제안한 물체 추적 방법과 기존의 방식인 객체 추적과 객체 인식과의 성능을 비교이다. 첫번째 그래프 1은 객체 검출, 객체 추적, 제안하는 추적기의 속도를 비교하였다.

<표 1> 속도 비교



초록색선인 객체 검출은 가장 높은 시간을 보여줬다. 또한 파란색선인 객체 추적은 제안한 추적기 방식보다 좀 더 높은 시간을 보여줬다. 얼굴 검출은 7.10 초로 가장 많은 시간이 걸렸고, 다음으로 객체 추적은 1.52 초 그리고 마지막으로 제안한 추적기는 0.57 초로 최단 시간을 보여줬다. 이는 객체 검출 대비 최대 80.74%의 개선된 속도를 보여줬다.

<표 2> 정확도 비교



두번째 그래프 2에서는 정확도를 비교 하였다. 객체 검출인 초록색 그래프는 그래프와 같이 겹침(Occlusion)이 발생한 구간에서는 찾지 못하여 정확도가 0%인것을 확인할 수 있다. 또한 객체 추적인 파란색 그래프에서는 특정 구간에서는 객체 검출보다 뛰어난 성능을 보여주지만 객체를 찾던 도중 다른 곳을 찾아 버리는 일이 발생해 특정 상황에서 좋지 않은 결과를 볼 수 있다. 그러나 위의 두 경우에 반해 제안한 추적기는 객체 검출을 먼저 시도해 보다 정확한 위치를 찾기 위해 시도한다. 그 후 겹침, 폐색 등 특정 구간에서 객체를 찾지 못하는 경우 전 영상에서 저장된 얼굴 데이터를 이용하여 객체 추적을 시도한다. 위의 그래프에서 나오는 것과 같이 제안하는 추적기는 빨간색 그래프이다. 빨간색 그래프는 위의 두 경우와 다르게 좀더 높은 정확도를 갖고 있는 것을 확인할 수 있었다.

6. 결론

최종 결론으로 제안하는 추적기는 속도와 정확도 모두 기존 연구 대비 우수한 성능을 보였다. 속도 부분에서는 기존 연구 대비 최대 80.74%의 성능 개선을 보여줬다. 또한 정확도 부분에서 최대 86.402%의 정확도를 보여 객체 추적의 85.103%, 객체 검출의 80.855%보다 높은 정확도를 보였다. 얼굴을 추적하기 매우 힘든 구간에서 객체 추적을 사용해 성능 개선을 확인하였다.

참고문헌

[1] Yang, Ming-Hsuan, David J. Kriegman, and Narendra Ahuja, "Detecting faces in images: A survey." IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 1, pp. 34-58, 2002.
 [2] Viola, Paul, and Michael J. Jones. "Robust real-time face detection," Int'l Journal of Computer Vision, vol. 57, no. 2, pp. 137-154, 2004.
 [3] COMANICIU, D., RAMESH, V., ANDMEER, P. "Kernel-based object tracking." IEEE Trans. Pattern Analysis

- and Machine Intelligence. 564-75. 2003. May.
- [4] COMANICIU, D., RAMESH, V., AND MEER, P. "Real-Time Tracking of Non-Rigid Objects using Mean Shift", In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 142-149. 2000. Jun.
- [5] Raul Viola, Michael J. Jones, "Robust Real-Time Face Detection", International Journal of Computer Vision. Volume 57, Issue 2, pp. 137-154. 2004. May.
- [6] Online Object Tracking: A Benchmark Yi Wu, Jongwoo Lim, Ming-Hsuan Yang; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 2411-2418
- [7] Ojala, Timo, Matti Pietikainen, and David Harwood. "A comparative study of texture measures with classification based on featured distributions," Pattern Recognition vol. 29, no. 1, pp. 51-59, 1995.
- [8] HUTTENLOCHER, D., NOH, J., AND RUCKLIDGE, W. "Tracking nonrigid objects in complex scenes.", In IEEE International Conference on Computer Vision (ICCV). 93-101. 1993.
- [9] C. Stauffer, E. Grimson, "Learning pattern of activity using real-time tracking", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.22, no.8, pp. 747-757, 2000
- [10] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," ACM Computing Surveys, vol. 38, no. 4, pp. 1-45, 2006.
- [11] Ojala, T., Pietikainen, M. and Harwood, D., A Comparative Study of Texture Measures with Classification Based on Feature Distributions. Pattern Recognition 29(1):51-59, 1996.
- [12] T. Ahonen, A. Hadid, and M. Pietikinen, "Face description with local binary patterns: Application to face recognition," PAMI 2006.
- [13] S. Liao, X. Zhu, Z. Lei, L. Zhang and S. Z. Li, "Learning Multi-scale Block Local Binary Patterns for Face Recognition," ICB 2007.