

가려짐에 강인한 실시간 얼굴추적 방법

*이준환, **정현조, ***유지상
광운대학교 전자공학과

*tarje3@kw.ac.kr, **guswh7905@kw.ac.kr, ***jsyoo@kw.ac.kr

Real-time Face Tracking Method Robust to Occlusion

*Jun-Hwan Lee, **Hyun-Jo Jung, ***Ji-Sang Yoo
Department of Electronic Engineering, Kwangwoon University

요 약

본 논문에서는 실시간 얼굴 추적을 위하여 기존의 CamShift 알고리즘의 단점을 보완한 새로운 CamShift 알고리즘을 제안한다. 배경 내 추적 객체와 색상이 유사한 객체가 존재할 경우 기존 CamShift 알고리즘은 불안정한 추적을 보여준다. 이러한 문제점을 화소 단위로 거리정보를 획득할 수 있는 Kinect의 깊이 정보와 HSV 색공간 기반의 피부색 후보영역을 추출하는 Skin Detection 알고리즘을 이용하여 색상분포만 이용하는 기존의 CamShift의 단점을 보완한다. 또한 추적하던 객체가 사라지거나 가려짐이 발생할 경우에도 다시 추적할 수 있는 특징점 기반의 매칭 알고리즘을 통하여 차폐영역에 강인한 특성을 가지게 한다. 이러한 향상된 CamShift 알고리즘을 사람의 얼굴 추적에 적용함으로써 다양한 분야에 활용 가능한 강인한 얼굴추적 알고리즘을 제안하고자 한다. 실험결과 제안하는 알고리즘은 기존의 추적 알고리즘인 TLD 보다 월등히 빠른 처리속도와 더 우수한 추적성능을 보여주었고, CamShift 보다 조금 느리지만 기존의 CamShift가 가지고 있는 문제점들을 해결하였다.

1. 서론

실시간 영상에서 사람의 얼굴을 검출하는 연구는 꾸준히 연구되고 있는 주제이다. 단조로운 배경이 아닌 동적 환경에서 얼굴을 추적하기 위한 연구가 활발히 진행 되어왔다. 하지만, 복잡한 배경과 동적인 환경에서 실시간으로 높은 정확도의 얼굴 추적은 여전히 연구과제로 남아있다. 실시간 얼굴 추적은 피부색, 움직임 정보, 다양한 특징 변화 등 고려 사항이 많기 때문에 쉬운 문제가 아니다. 정확하게 얼굴 영역을 찾았다고 해도 추적을 하려면 시간이 오래 걸리거나 또는 추적에 실패하는 경우도 있다. 이러한 경우는 실시간 얼굴추적이 필요한 HCI(Human Computer Interaction) 시스템이나 감시 시스템에 적합하지 않다. 실시간으로 얼굴 추적이 가능하기 위해서는 알고리즘의 계산량이 굉장히 중요하다. 아무리 정확한 얼굴 추적기라도 계산량이 많아 속도가 느리다면 활용할 수 있는 분야에 제약이 있기 마련이다. 뛰어난 추적성능을 자랑하는 Face-TLD[9] 같은 경우에도 계산량이 많아 초당 5 프레임 이하의 처리속도를 보여준다.

얼굴 추적을 위해 필요한 선행 작업은 얼굴영역을 검출하고 크기와 위치를 구하는 것이다. 실시간으로 얼굴을 검출하는 작업은 빠른 연산이 요구되기 때문에 기존의 연구에서는 주로 특징 기반 방법으로 색상정보와 얼굴형태 정보를 이용한 방법들이 제시 되어왔다. 색상정보를 이용한 방법은 구현이 쉽고 계산량이 적다는 장점이 있으나 입력된 영상이 유사한 컬러 성분으로 존재 할 경우 검출의 정확도가 낮아지는 문제점이 있다. 또한 얼굴형태 정보를 이용하는

방법은 조명변화에는 강건하지만 유사한 형태를 갖는 객체에 대해서 쉽게 오류를 범하는 경우가 있다. 본 논문에서는 획득된 동영상에서 실시간으로 얼굴영역을 검출하고 추적하는 새로운 알고리즘을 제안하고자 한다. 얼굴 영역의 검출은 색상정보를 이용하여 얼굴의 후보영역을 생성한 뒤 Haar feature 를 이용하여 특징 기반[1]으로 얼굴을 검출한다. 얼굴 검출 시 Haar feature 기반의 Adaboost 강분류기가 줄무늬나 글자와 같은 영역을 얼굴로 오검출하는 경우가 가끔 발생하는데 피부색의 색상정보[4]를 이용하여 얼굴의 후보영역을 생성한 뒤 얼굴을 검출하게 되면 오검출의 빈도수가 확연하게 줄어든다.

검출된 얼굴을 추적하기 위해서는 색상정보를 이용한 추적 방법인 CamShift 알고리즘[3]을 사용한다. CamShift 알고리즘은 목표 모델과 후보군 사이의 색상 분포인 히스토그램의 유사도를 비교하는 탐색 알고리즘으로서, 계산의 단순성, 안정성 면에서 좋은 성능을 가지고 있다. 그러나 목표모델과 유사한 색상분포를 갖는 객체나 배경에 취약하다는 단점이 있다. CamShift 알고리즘은 가변적인 탐색 윈도우를 사용하는데, 이는 추적하던 객체의 크기가 변해도 추적할 수 있는 장점이기도 하지만, 유사한 색상분포를 갖는 배경과 추적하던 객체가 근접하면 배경 전체를 추적하던 객체로 인식해 잘못된 추적을 하게 되는 문제점도 발생시킨다. 따라서 얼굴영역을 정확히 검출하여도 추적하는 과정에서 실패하는 경우가 발생할 수 있다.

얼굴을 추적하는 과정에서 색상정보 이외에 깊이 정보 등의 추가적인 정보를 사용하여 성능을 향상시킬 수 있다. 기본적으로 깊이 정보를 얻는 방법은 다수의 카메라에서

영상을 획득한 후 각 화소의 시차 값을 계산하는 방식으로 이루어진다. 하지만 여러 대의 카메라를 캘리브레이션 하는 과정에서 상당한 계산 시간을 요구하기 때문에 얼굴추적과 같은 실시간 추적 알고리즘에서는 적합하지 못하다. Microsoft 사의 Kinect 는 일반 카메라의 RGB 컬러카메라 기능과 적외선 센서를 이용하여 3 차원의 깊이 정보를 실시간으로 획득하는 기능이 동시에 가능하다. 본 논문에서 제안하는 알고리즘은 Kinect 를 통해 화소 단위의 거리정보를 획득한 후 이를 이용하여 기존 Camshift 의 단점을 개선한 향상된 추적 알고리즘이다. 또한 추적하던 얼굴이 고속으로 이동하거나 차폐영역 즉, 가려짐이 발생하여 추적이 실패하였을 경우에도 이전 얼굴 추적 시 저장해두었던 얼굴의 템플릿과 현재 프레임간의 특징점 기반 매칭을 수행함으로써 재추적이 가능하다.

본 논문의 구성은 다음과 같다. 2 장에서는 깊이 정보를 사용하여 캡슈프트 기반의 실시간 얼굴추적 방법을 새로이 제안한다. 3 장에서는 실험을 통해 제안하는 알고리즘과 기존의 알고리즘의 성능 비교를 제시하고 4 장에서 결론을 맺는다.

2. 가려짐에 강인한 CamShift 기반 실시간 얼굴추적 방법

기존의 CamShift 알고리즘은 객체 추적 등의 다양한 영상처리에 활용되고 있으나, 아직까지 개선의 여지가 있다. CamShift 알고리즘은 초기 탐색 윈도우 설정 문제, 가변적인 탐색윈도우의 잘못된 추적 문제, 탐색 윈도우를 벗어난 대상 객체의 추적 문제 등을 가지고 있다. 본 논문에서 제안하는 알고리즘은 이 세 가지 문제를 개선한 향상된 추적 알고리즘이다. CamShift 알고리즘을 수행하기 위해서는 사용자에게 의해 초기 탐색 윈도우의 위치 및 크기가 설정되어야 한다. 추적할 대상의 영역 설정은 불가피하지만 알고리즘을 반복적으로 수행하거나 실제 시스템에 적용할 경우 탐색 윈도우를 매번 설정해야 하는 번거로움이 있다. 추적을 시작할 때마다 초기 탐색 윈도우를 설정해야 하는 문제는 Haar feature 를 이용한 face detection 으로 해결한다. 영상에서 얼굴영역을 찾고 그 위치를 자동으로 초기 탐색 윈도우로 설정함으로써 사용자가 매번 탐색 윈도우를 설정해야 하는 번거로움을 덜 수 있다.

그림 1(a)는 CamShift 를 이용한 추적 과정 중 초기 탐색윈도우로 지정된 객체의 색상정보 분포를 계산하여 추적 대상과의 색상 유사확률을 0 부터 255 까지의 밝기 값으로 시각화한 Back-projection 영상이다[3]. 그림 1(b)는 이전프레임에서 추적된 얼굴영역의 중심점과 같은 깊이를 가지는 화소만을 255 의 밝기 값으로 시각화한 그림이다. 그림 1(c)는 그림 1(a) 와 그림 1(b)의 교집합 즉, AND 연산의 결과이다. 그림 1(d)는 입력영상인 RGB 컬러영상에서 추적의 결과를 빨간색 타원으로 표시한 영상이다. 기존의 CamShift 알고리즘에서는 그림 1(a)와 같은 Back-projection 영상을 사용하여 색상기반의 추적을 한다. 반면에 본 논문에서 제안하는 개선된 CamShift 알고리즘에서는 그림 1(c)에 주어진 깊이 정보를 사용하여 색상기반의 추적을 함으로서 유사색상을 가진 배경에 강인한 특성을 가지게 된다.

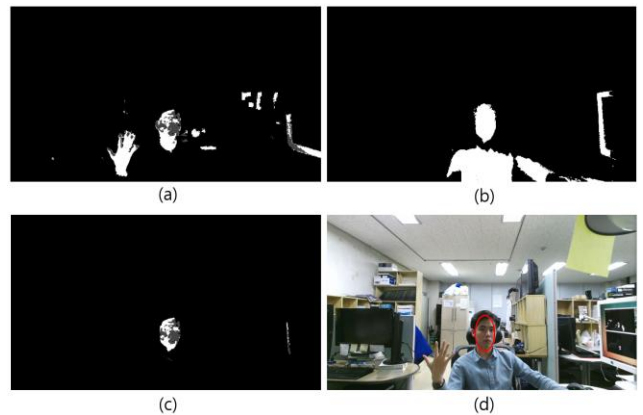


그림 1. (a) CamShift 의 Backprojection (b) 이전 프레임 추적결과 중심점의 깊이와 같은 깊이를 가지는 화소 (c) (a)와 (b)의 AND 연산 결과 (d) 얼굴 추적의 결과

객체의 고속 이동이나 가려짐으로 인해 추적에 실패한 경우 선별되어 저장된 템플릿 이미지들과 현재 프레임간의 특징점 매칭을 수행하여 현재 프레임에서 추적할 객체를 다시 찾는다. 특징점을 추출하는 방법과 추출된 특징점을 기술하는 방법은 다양하다. 본 논문에서 제안하는 알고리즘은 실시간으로 적용되어야 하기 때문에 연산 속도에 우선 순위를 두고 특징점 추출 알고리즘과 기술자를 선택한다. 특징점을 추출하는 알고리즘은 속도가 빠르고 반복성(Repeatability)이 뛰어난 FAST 알고리즘[5]을 적용한다. 특징점의 기술자에서 SIFT 알고리즘[7]의 경우는 128 차원으로 가장 자세하게 기술되어 높은 정확도를 기대할 수 있지만, 기술자의 차수가 높은 만큼 계산량이 많아지기 때문에 제안하는 알고리즘처럼 실시간 기반의 시스템에서는 적합하지 않다. 따라서 제안하는 알고리즘에서는 속도에 큰 강점을 가지고 있는 BRIEF 기술자[6]를 사용하여 특징점 매칭을 수행한다.

그림 2 은 차폐영역에 의해 객체가 가려져 추적이 중단된 후 객체를 재추적 하는 경우이다. 그림 2 에서 왼쪽 위의 작은 영역은 저장되어있는 템플릿 중의 하나이고 오른쪽 영역은 현재 프레임이다. 그림 2(a)는 저장되어있는 여러 템플릿들과 특징점 매칭을 수행하고 있는 영상이고 그림 2(b)는 특징점 매칭 후 호모그래피 계산을 통해 성공적으로 추적하던 객체를 다시 찾은 영상이다.



그림 2. (a) 여러 템플릿들과의 특징점 매칭 결과 (b) 호모그래피 변환으로 다시 찾은 추적 객체

3. 실험결과

제안하는 알고리즘은 Microsoft 사의 Kinect 를 통해

3 채널 1920x1080 해상도의 RGB 영상과 1 채널 512x424 해상도의 깊이 영상을 입력 받아 RGB 영상의 해상도를 960x540 로 리사이즈 한 후 그래픽카드(GPU)의 고속화 없이 Intel i5-4690 3.50GHz CPU, 16GB RAM, Visual Studio 2013 환경에서 실험하였다. 제안하는 알고리즘의 성능은 Face-TLD: Tracking-Learning-Detection [9], CamShift [3] 와 비교하였다. 실험영상은 총 726 프레임으로 구성되어 있는 자체 촬영한 시퀀스를 이용한다.

그림 3 은 비교하는 세 알고리즘의 처리속도를 비교한 그림이다. 비교하는 세 알고리즘 모두 처리속도가 0FPS 부터 시작하는데 추적할 객체의 초기값을 설정하기 전이므로 최초 0FPS 속도를 갖는 몇 개의 프레임은 최소 FPS 와 평균 FPS 계산 시 제외하였다. TLD 알고리즘은 얼굴이 가려져 있어 얼굴영역을 추출할 수 없는 상황에서 처리속도가 빨라지는 특징을 보였다. 반면에 제안하는 알고리즘은 얼굴이 가려져 영역을 찾을 수 없는 상황에서 처리속도가 저하되는 특징이 있다. 이는 일반적인 상황에서는 깊이 정보를 사용한 CamShift 알고리즘으로 처리속도가 상대적으로 빠르지만 객체를 찾을 수 없는 상황에서는 저장된 얼굴 템플릿과 현재 프레임간의 특징점에 기반한 매칭 작업이 많은 연산을 요구하기 때문이다. CamShift 알고리즘은 최대 30FPS, 최소 28FPS 의 속도로 빠르고 안정적인 속도를 보여주지만 추적에 실패한 뒤 재추적 하지 못하는 문제점으로 인해 349 프레임 이후에는 그림 3 에서와 같이 처리속도가 존재하지 않는다.

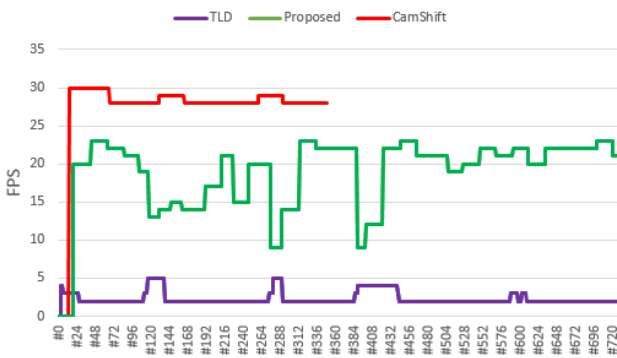


그림 3. 세 알고리즘의 처리속도 비교

표 1 은 TLD, CamShift, 그리고 제안하는 알고리즘에 대한 성능 비교 결과를 정리한 것이다. 속도 면에서는 CamShift 알고리즘, 제안하는 알고리즘, TLD 알고리즘 순으로 좋은 성능을 보여주었고 정확도 면에서는 제안하는 알고리즘, TLD 알고리즘, CamShift 알고리즘 순으로 좋은 성능을 보여주었다. 세 알고리즘 중 CamShift 알고리즘은 재추적이 불가능한 치명적인 문제점으로 인해 추적 알고리즘으로는 부적합하다는 판단이다. TLD 알고리즘은 처리속도가 평균 2.362 FPS 로 GPU 를 사용하여도 실시간으로 구현된다는 보장이 없다. 반면에 제안하는 알고리즘은 평균 19.336 FPS 로 GPU 을 사용하여 고속화 한다면 실시간 구현에 문제가 없어 보인다.

Table 1. 알고리즘 성능 비교

	TLD Algorithm	Proposed Algorithm	CamShift Algorithm
Max FPS	5 FPS	23 FPS	30 FPS
Min FPS	2 FPS	9 FPS	28 FPS
Average FPS	2.362 FPS	19.336 FPS	28.500 FPS
Object Miss Frame Count	61 Frame	43 Frame	377 Frame
Total Frame Count	726 Frame	726 Frame	726 Frame
Tracking Success rate	91.597 %	94.077 %	48.071 %

4. 결론

본 논문에서는 Microsoft 사의 Kinect 로부터 획득된 깊이 정보를 사용하여 기존 CamShift 의 단점을 개선한 얼굴객체 추적 알고리즘을 제안하였다. 제안한 알고리즘에서는 CamShift 가 가지고 있는 유사한 색상의 객체가 인접했을 때의 잘못된 추적, 추적 실패 시 재추적 하지 못하는 문제, 추적할 대상을 사용자가 직접 입력해야 하는 문제 등을 해결하였다. 기존에도 얼굴 추적에 대한 연구는 많이 이루어졌지만 뛰어난 정확도와 성능을 보이는 알고리즘들은 대체로 연산량이 많고 연산 시간이 오래 걸려 실시간 구현이 어려운 문제가 있다. 제안하는 알고리즘은 그래픽카드(GPU)의 고속화 없이도 Intel i5-4690 3.50GHz CPU, 16GB RAM, Visual Studio 2013 환경에서 평균 19FPS 정도의 높은 처리 속도를 보여준다. 그림 3 과 같이 객체가 사라지는 경우를 제외하면 20~25FPS 의 평균 속도를 보여주며 얼굴 추적 성공률도 기존의 알고리즘들 보다 우수함을 알 수 있다.

ACKNOWLEDGEMENT

이 논문은 2016 년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No. R0126-16-1034, 채널/객체 융합형 하이브리드 오디오 콘텐츠 제작 및 재생기술 개발)

5. 참고문헌

[1] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. vol. 1, pp. 511, 2001

[2] <https://www.flickr.com/photos/unavoidablegrain/6884354772/in/photostream/> (Image by Greg Borenstein)

[3] Bradski, Gary R. "Computer vision face tracking for use in a perceptual user interface." (1998).

- [4] Vezhnevets, Vladimir, Vassili Sazonov, and Alla Andreeva. "A survey on pixel-based skin color detection techniques." Proc. Graphicon. vol. 3, pp.85-92, September, 2003.
- [5] Rosten, Edward, and Tom Drummond. "Machine learning for high-speed corner detection." Computer Vision- ECCV 2006. Springer Berlin Heidelberg. pp. 430-443, 2006.
- [6] Calonder, Michael, et al. "Brief: Binary robust independent elementary features." Computer Vision- ECCV 2010 pp. 778-792, 2010.
- [7] Lowe, David G. "Distinctive image features from scale-invariant keypoints." International journal of computer vision vol. 60, no. 2, pp. 91-110, 2004.
- [8] Kalal, Zdenek, Krystian Mikolajczyk, and Jiri Matas. "Tracking-learning-detection." Pattern Analysis and Machine Intelligence, IEEE Transactions on vol. 34, no. 7, pp. 1409-1422, 2012.
- [9] Kalal, Zdenek, Krystian Mikolajczyk, and Jiri Matas. "Face-tld: Tracking-learning-detection applied to faces." Image Processing (ICIP), 2010 17th IEEE International Conference on. IEEE, 2010.
- [10] Bay, Herbert, et al. "Speeded-up robust features (SURF)." Computer vision and image understanding vol. 110, no. 3, pp. 346-359, 2008.
- [11] Harris, Chris, and Mike Stephens. "A combined corner and edge detector." Alvey vision conference. vol. 15. 1988.
- [12] Trzcinski, Tomasz, and Vincent Lepetit. "Efficient discriminative projections for compact binary descriptors." Computer Vision- ECCV 2012. Springer Berlin Heidelberg. pp. 228-242, 2012.