

깊이정보와 컬러정보를 이용한 빠른 다중 얼굴 검출 및 추적 알고리즘

*김우열 **배윤진 ***서영호 ****김동욱

광운대학교

*wykim@kw.ac.kr

Fast Multiple Face Detection and Tracking Algorithm using Depth and Color Information

*Kim, Woo-Youl **Bae, Yun-Jin ***Seo, Young-Ho ****Kim, Dong-Wook

Kwang-woon University

요약

본 논문에서는 컬러영상과 깊이영상을 이용하여 여러 명의 얼굴을 검출하고 추적하는 알고리즘을 제안한다. 제안하는 알고리즘은 얼굴 검출부와 추적부로 나뉘어져 있으며, 얼굴 검출 방법은 기존의 Adaboost를 이용하지만, 속도 개선을 위해 깊이정보와 컬러정보를 이용하여 탐색영역을 얼굴이 존재하는 영역으로 제한하여 얼굴을 검출한다. 얼굴 추적 방법은 템플릿 매칭 방법과 나선형 탐색방법을 사용하며, 그리고 조기 종료 기법을 사용하여 수행시간을 줄였다.

1. 서론

인간생체의 일부를 검출하고 추적하는 방법은 컴퓨터 비전분야를 비롯한 다양한 분야에서 오래전부터 연구되어 왔다. 주된 사용 범위로 는 보안시스템 화상회의, 로봇비전, HCI에 의한 대화형 시스템, 스마트 홈 등에 널리 사용되고 있다. 이들 중 얼굴에 대한 연구가 가장 활발히 진행되어 왔으며, 빠르고 정확한 얼굴 검출과 추적에 목적을 두고 있다.

얼굴 검출을 위한 연구들은 크게 지식 기반(Knowledge-Based), 특징 불변 방법(Feature invariant approaches), 템플릿 기반(Template-Based), 외형 기반(Appearance-Based)방법으로 나눌 수 있다[1]. 지식 기반 방식은 얼굴의 구성 요소인 눈, 코, 입 등에 존재하고, 그 요소들은 일정한 거리 관계를 가진다는 가정을 이용하여 얼굴을 검출하는 방법이다[2]. 특징 불변 방법은 얼굴에 변하지 않는 특징(피부색, 질감, 얼굴요소)을 이용하여 얼굴을 검출한다[3][4]. 템플릿 기반 방식은 수동적으로 미리 정의하거나 규칙에 의해 생성된 표준 얼굴 패턴을 생성하여 이를 입력영상과 비교하여 얼굴을 검출하는 방법이다 [5]. 마지막으로 외형 기반 방법은 템플릿 기반 방법과는 달리 미리 정의한 표준 패턴을 사용하는 것이 아니라 얼굴과 비 얼굴 영상으로 구성된 다양한 학습 영상 집합을 입력받아 훈련과정을 통해 얼굴과 비 얼굴을 분류해 낼 수 있는 분류기를 생성하고 이를 이용하여 영상에서 얼굴을 찾아내는 방법이다.

얼굴 추적은 동영상으로 입력되는 영상 시퀀스에서 움직이는 사람의 얼굴을 검출하여 이동 경로를 추적하는 것으로 실시간 환경에서의 빠른 수행속도에 초점을 맞추어 연구가 진행되고 있다. 얼굴을 추적하는 방법에는 기존에 2차원 영상을 사용하던 방법과는 달리 3차원적 정보의 깊이 값을 이용하는 방법들도 연구가 진행되고 있다. 또한 최근

에는 깊이카메라 또는 Microsoft사의 Kinect를 이용하여 깊이정보를 실시간으로 획득하여 얼굴 검출 및 추적에 직접 사용하는 연구도 진행되고 있다.

본 연구에서는 여러 명의 얼굴을 빠르고 정확하게 검출 및 추적하는 것이 목적이다. 본 연구에서는 깊이정보와 컬러정보를 사용하며, 깊이정보는 깊이카메라로(Kinect)부터 획득하였다. 본 논문에서 제안하는 시스템은 얼굴 템플릿을 검출하는 검출부와 검출된 템플릿을 추적하는 추적부 두 부분으로 구성되어 있다.

본 논문의 2장에서는 제안하는 알고리즘에 대하여 설명하고, 3장에서는 실험 결과를 기술하고, 4장에서 결론을 맺는다.

2. 제안하는 얼굴 검출 및 추적 알고리즘

제안하는 얼굴 검출 및 추적 알고리즘의 흐름도는 다음과 같다. 제안하는 과정은 얼굴 검출부와 추적부로 구성되어 있으며 얼굴 검출과정은 기본적으로 첫 프레임만 수행한다. 검출과정에서 검출된 얼굴영역은 얼굴 추적과정에서 템플릿으로 사용되며 얼굴이 검출된 다음 프레임부터는 추적과정만 수행된다. 그러나 장면이 바뀌거나 추적과정에서 템플릿 매칭이 이루어지지 않은 경우 다시 검출과정을 수행한다.

가. 얼굴 검출 알고리즘

본 논문의 얼굴 검출 과정에서는 Adaboost 알고리즘[6]을 사용하지만 Adaboost 알고리즘의 입력 영상의 크기를 줄여 연산시간을 줄였다. 본 연구에서는 입력영상을 제한하기 위하여 깊이정보와 컬러정보를 이용하였다. 먼저, 그림1의 입력된 깊이 동영상의 차영상(a)을 구하여 움직임을 검출한다. 차영상은 이전 프레임과 현재프레임과의 차를 통하여 영상을 획득한다. 본 연구에서는 움직임에 대한 실험과 경험에

의하여 이전프레임과 현재프레임과의 차를 통하여 영상을 획득하였다. 움직임이 검출되지 않으면 컬러정보의 피부색 검출을 이용하여 얼굴 영역을 정의한다[7]. 움직임이 검출되어 차영상이 구해지면, 모폴로지(morphology) 필터를 이용하여 잡음을 제거한다(b). 필터링한 차영상을 가지고 수직, 수평방향으로 누적 덧셈을 하여 움직임 영역을 정의한다(c). 그리고 컬러영상에서 깊이영상에서 구한 움직임 영역(그림1.(c)에서의 빨간색 박스 영역)에 대하여 피부색을 검출한다(d). 차영상과 같은 방법으로 모폴로지(morphology) 필터를 이용하여 잡음을 제거한 다음 얼굴영역을 정의한다(e). 정의된 얼굴영역은 Adaboost의 입력영상으로 들어가게 되어 얼굴을 검출한다(f). 검출된 얼굴의 밝기 성분을 얼굴 추적 과정에서 템플릿으로 사용하고 해당 깊이영상을 보조 데이터로 사용한다. 또한 템플릿의 좌표 값도 추적과정으로 보낸다.

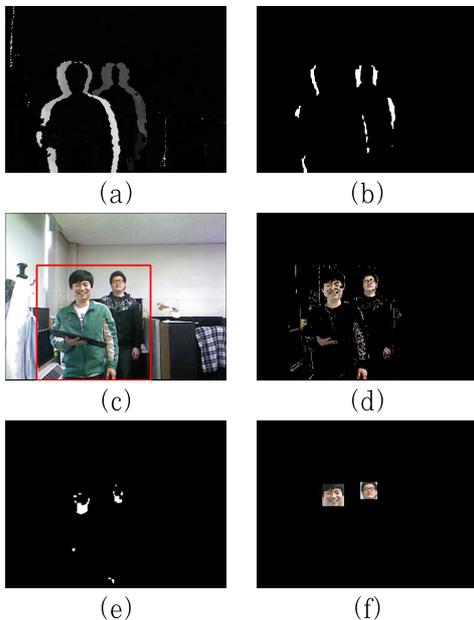


그림1. 단계별 얼굴 검출 과정 영상 (a)깊이 차영상, (b)필터링 된 깊이 차영상, (c)움직임 영역 (d)움직임 영역에서의 피부색 검출, (e)필터링 된 피부색 검출, (f)Adaboost 알고리즘을 이용하여 검출된 얼굴영역

나. 얼굴 추적 알고리즘

얼굴추적 과정은 얼굴검출 과정에서 생성되거나 그 전 프레임에서 갱신된 템플릿이 현재 영상에서 매칭 되는 지점을 찾는 과정이다. 그러나 전체 프레임을 대상으로 템플릿 매칭을 수행하면 과도한 시간이 소요된다. 따라서 본 논문에서는 탐색영역을 최소화하는 방법을 제안하며, 얼굴이 상하좌우 뿐만 아니라 앞뒤로 움직이는 경우를 포함하도록 하였다.

얼굴추적을 위해 탐색하여야 하는 영역은 얼굴의 움직임 속도와 관련이 있다. 본 논문에서는 추적할 얼굴이 전방의 영상을 시청하고 있다고 가정한다. 얼굴의 움직임에 대해서는 시청하면서 움직일 수 있는 최대의 움직임 속도를 고려하여야 하며, 깊이에 반비례하여 얼굴의 움직임 양이 결정된다는 것을 고려하여야 한다. 거리에 따른 물체의 크기는 동일한 물체가 z_1 과 z_2 에 있을 때의 크기를 각각 s_1 과 s_2 라 할 때 식 (1)과 같은 관계를 갖는다.

$$s_2 = \frac{z_2}{z_1} s_1 \quad (1)$$

따라서 특정 깊이에서의 크기를 알면 그 물체가 깊이이동을 하였을 때의 크기를 식 (1)으로 쉽게 구할 수 있다. 물체의 깊이에 따라 움직임 속도, 즉 움직임 양이 달라지므로 깊이에 따른 움직임 양을 일반적으로 표현하기 위해서 본 논문에서는 현재의 템플릿에 대비한 상대적인 크기로 탐색영역을 정의한다. 본 논문에서는 먼저 얼굴의 최대 움직임 속도를 실험을 통하여 전후와 좌우방향에 대해 측정하였으며, 이를 두 프레임 간의 거리를 환산하여 탐색영역을 정의하였다.

추적과정에서 얼굴이 전후로 움직일 때 깊이가 변화하기 때문에 현재의 템플릿을 사용할 경우 정확한 추적을 할 수 없을 뿐만 아니라 깊이가 변화함에 따라 탐색범위도 변화시켜야 한다. 그러나 이를 위해서는 전후로 얼굴이 움직인 양을 측정하여야 한다. 본 논문에서는 추적 대상인 현재 프레임에서 현재 템플릿과 동일한 위치와 동일한 크기의 영상을 샘플링하여 사용한다. 측정방법은 템플릿과 샘플영상을 각각 $m \times n$ 서브블록으로 나누고, 각 서브블록 (i, j) 의 깊이 값 평균 $(a_{i,j})$ 을 계산하여 그 중 최고치를 각각 템플릿(z_{temp})과 샘플링된 현재 프레임(z_{cur})의 깊이 값으로 선택한다. 이것을 식 (2)에 나타내었다. 여기서 첨자 $temp$ 와 cur 은 각각 템플릿과 샘플링된 현재 프레임을 표시하며, $\max()$ 는 괄호내의 값 중 최대치를 선택한다.

$$z_{temp} = \max(a_{1,1}^{temp}, \dots, a_{n,m}^{temp}), z_{cur} = \max(a_{1,1}^{current}, \dots, a_{n,m}^{current}) \quad (2)$$

얼굴추적의 다음 단계는 앞 절에서 재조정된 템플릿을 사용, 재조정된 탐색영역을 조사하여 매칭 된 점을 찾는 것이며, 기본적으로는 탐색영역 전체를 탐색한다. 이 경우 탐색하여야 하는 화소 수는 $(s_h^{search} - s_h^{temp}) \times (s_v^{search} - s_v^{temp})$ 이며, 여기서 s_h^{temp} 와 s_v^{temp} 는 각각 템플릿의 수평 및 수직방향 크기이며, 이에 해당하는 현재 프레임의 탐색범위의 값은 각각 s_h^{search} 과 s_v^{search} 이다. 본 논문에서는 탐색할 때 비용함수로 SAD(sum-of-absolute differences)를 사용한다. 즉, 탐색범위 내의 모든 위치를 탐색하여 그 중 가장 작은 SAD값을 갖는 위치를 선택한다.

그러나 본 논문의 목적이 고속추적이므로 이를 위한 방안을 제안한다. 일반적으로 위치를 추적하는 목적에 따라 조금이 오차도 없이 추적하는 경우보다는 어느 정도의 근사를 허용하는 경우가 많고, 또 정확한 위치를 찾는 것은 매우 많은 연산을 요한다. 또한 일적으로는 시청하면서 잘 움직이지 않으며, 움직인다고 해도 그 크기가 크기 않다. 또한 상하나 전후의 움직임보다 좌우방향의 움직임이 많다.

이런 점을 착안하여 본 논문에서는 조기종료(early termination) 기법을 사용한다. 이를 위해서 근사적으로 위치를 추적했다고 판단하여야 하는데, 본 논문에서 비용함수로 SAD를 사용하기 때문에 미리 정한 문턱치 SAD값 이하를 갖는 경우 조기종료를 시행한다. 이를 식으로 나타내면 식 (3)와 같다.

$$SAD_{i,j} < T_l \quad (3)$$

여기서 $SAD_{i,j}$ 는 위치 (i,j) 에서의 SAD값이며, T_h 는 미리 정한 문턱치이고 이것은 실험적으로 결정된다. 또한 탐색범위 내의 각 위치를 탐색하는 순서도 위에서 언급한 것에 부합하도록 탐색범위의 중심에서부터 나선형으로 탐색하도록 하였다.

앞에서 언급한 바와 같이 얼굴추적 과정을 수행하다가 템플릿 매칭에 실패하면 얼굴검출 과정으로 귀환하여 얼굴검출 과정부터 다시 수행한다. 이것은 장면이 바뀌거나 사람이 화면에서 사라지는 경우 등에 나타난다. 이와 같은 경우는 추적과정에서 SAD값이 매우 크게 나타나는데, 본 논문에서는 식 (4)과 같이 탐색범위 전체를 탐색한 후 각 위치에서의 SAD값의 최소값이 문턱치 T_h 보다 큰 경우로 결정한다.

여기서 $\min()$ 는 괄호내의 값 중 최소치를 선택한다.

$$\min(SAD_{i,j} \text{ for all pixel positions within the search range}) > T_h \quad (4)$$

아래 그림3은 위에서 제안한 추적 알고리즘을 통한 여러 명의 얼굴을 추적 할 때의 모습을 보여준다. (a)는 2명일 때, (b)는 3명일 때 추적할 때의 영상이다.



그림2. 다중 얼굴 추적 영상 (a)2명일 때의 영상 (b)3명일 때의 영상

3. 실험 및 고찰

본 논문에서 제안하는 방법의 성능을 평가하기 위하여 Intel Core i5 CPU의 6GB RAM의 하드웨어, Microsoft window7 운영 체제에서 Microsoft visual studio 2008과 OpenCV Library 2.1[8]을 이용하여 알고리즘을 구현하였다. 또한 컬러 영상과 깊이 영상은 Kinect으로부터 획득되며 해상도는 640x480이다.

알고리즘의 속도를 측정하기 위해 직접 제작한 여러 종류의 테스트 시퀀스를 이용하였다. 아래 표1은 각각 300frame에 대하여 좌우, 전후, 상하 움직임에 대한 시퀀스를 가지고 제안한 얼굴검출과 얼굴추적에 대한 평균 수행시간을 계산한 것이다. 그리고 좌우, 전후, 상하에 대한 평균시간을 계산하였다.

표1. 300frame에 대한 평균 수행시간 비교.

시퀀스	제안한 얼굴검출 방법		제안한 얼굴추적 방법	
	1명	2명	1명	2명
좌우 움직임	28.55ms	60.75ms	1.73ms	28.12ms
전후 움직임	26.78ms	46.20ms	3.18ms	27.32ms
상하 움직임	24.63ms	43.07ms	4.36ms	26.67ms
평균	26.65ms	50.01ms	3.09ms	27.37ms

표1에서 볼 수 있듯이, 제안하는 검출 방법의 속도는 1명일 때는 평균 26.65ms로 비교적 빠른 속도를 보였으나 2명일 때는 50.01ms일 때는 속도가 많이 느려지는 것을 알 수 있었다. 이는 2명일 때, 움직임에 대한 범위와 피부색에 대한 범위가 넓어져 Adaboost의 입력영상으로 들어가는 크기가 커지기 때문이라고 생각된다. 제안하는 얼굴추적 방법 또한 1명일 때는 3.09ms로 비교적 빠른 속도를 보였지만 2명일 때는 27.73ms로 느려지는 것을 알 수 있었다.

4. 결론 및 향후 과제

본 논문에서는 깊이정보와 컬러정보를 이용한 빠른 얼굴검출 및 추적 방법을 제안하였다. 빠른 검출 및 추적 속도를 위해 검출부에서는 깊이정보와 컬러정보를 이용하여 Adaboost의 입력영상의 크기를 제한하였으며, 추적부에서는 조기종료 기법을 사용하여 수행시간을 줄였다.

수행시간을 측정했을 경우, 1명일 때는 빠른 검출과 추적의 속도가 나왔으나, 2명일 때는 다소 느려지는 경향을 보였다. 이는 향후 여러 명의 얼굴을 검출과 추적을 할 때도 1명일 때와 비슷한 속도가 나올 수 있도록 하는 연구가 필요할 것이다.

감사의 글

본 연구는 지식경제부, 방송통신위원회 및 한국산업기술평가관리원의 산업원천기술개발사업(정보통신)의 일환으로 수행하였음. [KI002058, 대화형 디지털 홀로그램 통합서비스 시스템의 구현을 위한 신호처리 요소 기술 및 하드웨어 IP 개발]

참고문헌

- [1] C. Liu, "A Bayesian discriminating features method for face detection," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 25, 6, pp. 725-740, June, 2003
- [2] G. Yang and T.S Huang, "Human Face Detection in Complex Background," Pattern Recognition, vol. 27, no. 1, pp. 53-63, 1994.
- [3] K.C. Yow and R. Cipolla, "Feature-Based Human Face Detection," Image and Vision Computing, vol. 15, no. 9, pp. 713-735, 1997.
- [4] J. Yang and A. Waibel, "A Real-Time Face Tracker," Proc. Third Workshop Applications of Computer Vision, pp. 142-147, 1996.
- [5] I. Craw, D. Tock, and A. Bennett, "Finding Face Features," Proc. Second European Conf. Computer Vision, pp. 92-96, 1992.
- [6] P. Viola and M. Jones, "Robust Real-Time Face Detection," Computer Vision, Vol. 52, No. 2, pp. 137-154, 2004.
- [7] D. Chai, K. N. Ngan, "Locating Facial Region of a Head-and-Shoulders Color Image," Int'l Conf. Automatic Face and Gesture Recognition, pp. 124-129, April 1998.
- [8] <http://sourceforge.net/projects/opencvlibrary/files/>