

배경화면 변화를 이용한 객체의 윤곽점 검출

백주호*, 이창수, 오해석
송실대학교 컴퓨터학과
e-mail:shadow3697@hotmail.com

Object Boundary Point Detection Using Background Image Change

Ju-ho Back*, Chang-Soo Lee*, Hae-Seok Oh*
*Dept. of Computing, Graduate School, Soongsil University

요 약

인터넷 시대에 접어들면서 웹 카메라를 이용한 보안 시스템의 개발이 활발하다. 원격지에 설치된 카메라가 보내준 영상을 통하여 현재의 상황을 파악할 수 있으며, 적절한 조치를 웹을 통해 취할 수 있다. 본 논문에서는 카메라로부터 입력되어지는 입력영상과 배경영상의 차를 이용하여 움직임 검출하는 방법을 제안한다. 또한 배경영상은 시간에 따라 변화하기 때문에 변화된 시점부터 배경이미지 픽셀을 교체 해준다. 카메라에서 받아오는 영상을 배경영상과 입력영상으로 구분 한 다음 두 영상의 차를 구하여 영상의 변화점을 찾는다. 픽셀 검사는 모든 픽셀을 연산에 참여하는 방식을 탈피하여 일정한 간격을 두고 이미지의 픽셀을 검색하여 효율적인 객체의 윤곽점을 추출한다.

1. 서 론

인터넷 시대에 접어들면서 웹 카메라를 이용한 보안 시스템의 개발이 활발하다. 원격지의 카메라로부터 전송된 영상을 통하여 현재의 상황을 파악할 수 있으며, 적절한 조치를 웹을 통해 취할 수 있다. 이러한 웹 멀티미디어 보안 시스템은 교통현황 파악, 건설현장이나 상가매장의 모니터링, 무인 시설물 감시 등에 사용되고 있다. 그런데 활용영역이 확대되면서 영상의 해상도와 전송속도, 그리고 보안시스템의 핵심인 객체영역 인식, 영상 정보의 처리, 저장, 검색 기술 등의 연구가 요구 되고 있다[1,2].

실시간 영상에서 배경영상과 입력영상을 구분하여 움직인 객체를 인식하거나 검출하기 위해 차영상을 이용한 방법, 블록정합기법, 배경영상을 이용한 방법 등을 이용한 연구가 주로 이루어지고 있다. 그러나 이들 방법은 배경영상에서 발생하는 잡음이나 조명의 변화 등에 의해 불필요한 잡음이 검출되는 문제점을 가지고 있다. 따라서 보다 정확한 객체 추적이나

추출할 필요가 있다.

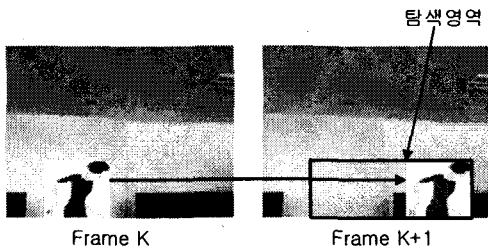
본 논문에서는 실시간 영상에서 초기의 배경영상을 기준으로 입력영상과의 차를 구하고 시간에 따라 변화하는 배경영상을 $N \times M$ 픽셀 마스크만큼 교체하여 갱신 한다. 이미지 픽셀 검사는 모든 픽셀을 연산에 참여시키지 않고 일정한 픽셀 간격을 두고 이미지의 픽셀을 검색하여 효과적으로 객체 유입시점부터 객체의 윤곽점을 추출하는 방법을 제안한다.

제안하는 방식을 사용한 객체의 윤곽점 검출방법은 기존의 윤곽점 검출을 사용할 때보다 조명이나 기존의 잡음에 강점을 가지고 있다. 따라서 실시간 영상영역 안으로 객체의 윤곽점을 정확히 검출할 수 있기 때문에 영상 보안시스템의 객체 추적이나 움직임 검출과 같은 분야에 적용이 가능하다.

본 논문에서 2장은 기존 연구방법을 분석하고, 3장은 배경영상의 변화와 픽셀 간격을 이용한 객체 윤곽점 검출방법을 제안한다. 4장에서는 제안한 방법으로 실험한 결과를 기술하고, 마지막으로 5장에서는 결론과 향후 연구 방향을 기술한다.

2. 관련 연구

이미지에서 객체의 윤곽점을 추출하는 기존의 방법으로는 블록 정합 기법은 현재프레임 탐색영역 안에서 이전프레임의 지정된 블록과 가장 유사한 블록을 찾는 방법이다(그림 1). 객체가 움직이지 않다가 다시 움직이는 경우에도 객체의 윤곽점을 추출하는 데에도 가능 하고 블록의 크기와 추적할 물체를 지정할 수 있다[2]. 그러나 탐색영역 밖으로 객체가 유입되었을 경우에는 초기의 블록과 유사한 블록으로 인식하지 못하는 단점을 가지고 있다[5].



(그림 1) 탐색영역 안에서 지정된 객체 탐색

차 영상을 이용한 방법은 현재 프레임에서 이전 프레임의 차를 구하여 객체를 실시간으로 추적하거나 추출 할 수 있다. 그러나 객체가 정지하는 경우는 객체를 배경으로 인식하는 단점을 가지고 있다.

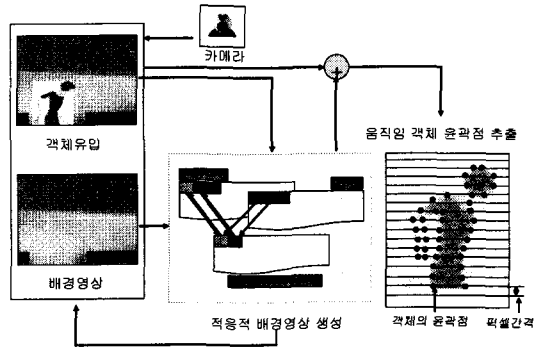
배경 영상방법은 프레임마다 현재 프레임에서 배경 영상을 빼는 방법이다. 객체에 대한 정확한 위치 정보와 형태 정보를 얻을 수 있다. 그러나 기상조건, 계절의 변화, 밝기의 영향 등 조그마한 환경 변화에 많은 영향을 받기 때문에 시간의 변화에 따른 정확한 배경보상을 필요로 한다[1,3,4].

3. 실시간 객체의 윤곽점 검출 방법

본 논문에서 제안하는 윤곽점 검출 방법은 카메라로부터 획득되는 초기 영상 이미지를 배경영상 이미지로 선택한다. 배경 영상 이미지는 객체의 유입이 없는 상태에서 획득된 순수한 배경이미지이다.

배경영상과 카메라로부터 입력되는 영상과의 차를 이용하여 다른 값이 존재하면 픽셀간격을 이용한 이미지 검사를 통하여 객체의 윤곽선에 해당하는 부분을 찾는다.

(그림 2)는 이러한 과정을 보이고 있다.



(그림 2) 객체의 윤곽점 검출과정

본 논문에서 제안하는 객체의 윤곽점 검출 과정은 다음과 같다.

- 단계 1. 카메라로부터 입력영상과 배경영상 획득
- 단계 2. 배경영상과 입력영상에서 밝기의 변화가 있는지 검사하고 변화가 있으면 배경영상을 픽셀단위로 입력영상의 배경에 해당하는 부분과 교체
- 단계 3. 배경영상과 입력영상의 차영상 획득
- 단계 4. 배경영상으로부터 픽셀 간격을 이용한 객체 윤곽점 추출

3.1 배경영상 획득 및 갱신

카메라의 입력 영상에 의해 얻어지는 이미지는 초기에 배경영상을 얻기 위해 물체의 변화나 이미지의 변화가 없는 것으로 간주한다.

배경영상은 카메라로부터 입력받은 후부터 시간이 지남에 따라 객체가 검출되지 않는 상황이더라도 배경영상 자체가 변하게 된다. 변화가 없는 공간에서 조명의 변화등과 같은 미세한 움직임에도 차 영상을 통하여 초기의 배경화면과 현재의 배경화면을 비교해보면 이미지 내에 잡음이 발생함을 할 수 있다. 이것은 시간이 지남에 따라 더 많은 잡음이 발생한다는 것이다. 따라서 객체의 유입이 없는 배경영상임에도 불구하고 객체의 유입이 있는 것과 같은 오류가 발생 한다.

본 논문에서는 이러한 잡음을 제거하여 보다 정확한 객체의 추출을 위해서 배경영상을 갱신하는 방법을 사용한다. 전체적인 배경영상의 갱신은 많은 연산량을 필요로 하기 때문에 객체의 윤곽점을 검출하는데 시간이 많이 소요된다. 따라서 $N \times M$ 마스크

를 사용하여 계속적으로 배경영상을 갱신하면서 객체의 윤곽점을 정확하게 판별할 수 있도록 한다.

적응적 배경영상 생성은 입력영상과 배경영상의 $N \times M$ 마스크 내에서 RGB 채널의 각각의 차이값을 구한다. 식(1)에서 RGB 채널 각각의 차이값이 임계값 α 보다 작으면, 객체가 아니라고 판단하고 입력영상에서 픽셀을 선택하여 배경영상을 갱신하여 준다. 임계값 α 보다 클 경우는 새로운 객체가 영상 안에 유입된 경우이므로 배경영상의 픽셀을 그대로 사용한다.

$$\begin{aligned}
 R_{Channel} &= abs(RValue(BackGround_{RImageMask}[i, j] \\
 &\quad - RValue(InputImage_{RImageMask}[i, j])) \\
 G_{Channel} &= abs(GValue(BackGround_{GImageMask}[i, j] \\
 &\quad - GValue(InputImage_{GImageMask}[i, j])) \\
 B_{Channel} &= abs(BValue(BackGround_{BImageMask}[i, j] \\
 &\quad - BValue(InputImage_{BImageMask}[i, j]))
 \end{aligned}$$

$$\begin{aligned}
 &if((R_{Channel} < \alpha) or (G_{Channel} < \alpha) or (B_{Channel} < \alpha)) \\
 &\quad Select InputImage[i, j] \\
 &else \\
 &\quad Select BackGroundImage[i, j] \quad (1)
 \end{aligned}$$

3.2 픽셀간격을 이용한 객체의 윤곽점 추출

실시간으로 배경영상으로 객체가 들어 왔는지 아닌지를 검사해야 한다. 우선적으로 배경영상과 입력영상과의 차를 이용하여 객체의 위치를 탐지 한다.

차 영상 방법은 이미지 내에 존재 하는 모든 픽셀을 검사하여 연산 시간을 증가 시켜 정지 영상에 해당하는 부분에는 가능하지만, 실시간으로 많은 양의 이미지를 검사 하는 방법에는 적절하지 않다. 그러므로 본 논문에서 제안하는 방식은 모든 픽셀을 연산에 참여 시키지 않고 일정한 픽셀 간격을 두고 객체의 윤곽점을 검출하게 된다.

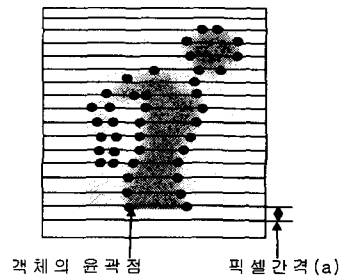
(그림 3)에서 이미지 검사는 실험적 경험을 바탕으로 일정한 간격을 설정해 주고 픽셀 간격만큼의 검사를 하게 된다.

픽셀 검사에서 배경영상과 입력영상의 R,G,B 픽셀간 차이가 임계값 β 값을 넘거나 같은 값을 가지면, 식(2)에 따라 움직인 객체가 있는 것으로 간주한다. 효율적인 움직임 검출을 위하여 미세한 객체의

움직임은 무시하기로 한다.

$$\begin{aligned}
 R_{Object} &= abs(RValue(BackGround_{RImage}[i, j] \\
 &\quad - RValue(InputImage_{RImage}[i, j])) \\
 G_{Object} &= abs(GValue(BackGround_{GImage}[i, j] \\
 &\quad - GValue(InputImage_{GImage}[i, j])) \\
 B_{Object} &= abs(BValue(BackGround_{BImage}[i, j] \\
 &\quad - BValue(InputImage_{BImage}[i, j]))
 \end{aligned}$$

$$\begin{aligned}
 &if((R_{Object} \geq \beta) or (G_{Object} \geq \beta) or (B_{Object} \geq \beta)) \\
 &\quad Object[i, j] = true \\
 &else \\
 &\quad Object[i, j] = false \quad (2)
 \end{aligned}$$



(그림 3) 픽셀 간격을 이용한 객체의 윤곽점 추출

4. 실험 결과 및 분석

4.1 실험 환경

본 실험은 명암도의 변화가 너무 크지 않은 오후 시간대를 정하여 웹 카메라를 사용하여 배경영상과 입력영상을 실시간으로 처리하여 실험하였다.

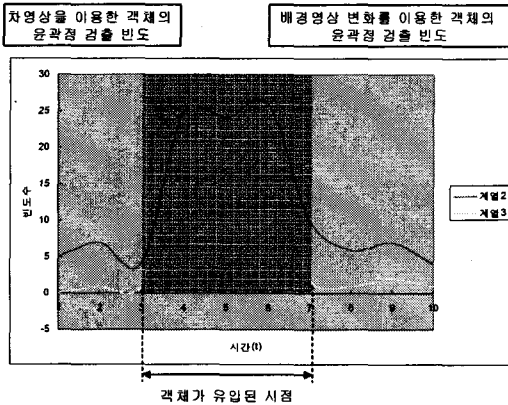
구현환경은 Intel Pentium 4 CPU 1.60GHz, 256M RAM의 PC에서 Visual C++ 6.0(Service Pack 5)을 이용하여 구현하였다. 배경영상과 입력영상은 화상 전송시를 고려하여 320×240의 RGB 24bit 칼라 영상을 이용하였다.

4.2 실험 결과

(그림 4)는 실시간 영상에 움직임 객체가 영상 안으로 들어왔을 때를 본 논문에서 제안한 배경화면

변화를 이용하여 객체의 윤곽점을 추출하였다.

실시간 영상 이미지를 사용하여 객체가 유입된 시점부터 객체가 이미지에서 사라지는 시점까지의 객체의 윤곽점을 추출하였다. 다음 그래프는 윤곽점이 추출되는 시점의 빈도수를 누적하여 측정하였다.



(그림 4) 객체의 윤곽점 검출 빈도

기존의 블록정합 방법은 객체를 블록 안에 있다는 가정 하에 움직임을 검출하고 차 영상만을 이용할 때는 잡음에 민감한 반응을 보이기 때문에 움직임 객체가 입력영상에 없다고 하더라도 움직임으로 검출하는 경우가 보였으나, 본 논문에서 제안한 방법은 움직임 객체가 있을 경우만을 검색하고, 객체를 검색하는 연산시간 또한 단축되었다.

5. 결론 및 향후 연구 방향

본 논문에서는 카메라로부터 입력되어지는 영상으로부터 실시간으로 유입되는 객체의 윤곽점을 검출하는 방법을 제안하였다.

실험환경은 웹 카메라를 이용하여 배경영상을 획득하고 실시간으로 입력되는 영상으로 유입된 객체의 윤곽점을 효과적 검출하였다.

객체의 윤곽점 검출은 보안 시스템에서 움직이는 객체를 추적하거나, 움직임 감지등으로 확대 연구 가능하다.

본 논문에서 제안한 방식의 객체의 윤곽점 검출방식은 배경영상 획득 및 갱신을 통하여 조명이나 미세한 움직임 변화에 의해 객체로 인식되는 오류를 개선하는데 효과적이다.

향후 연구 방향은 객체의 윤곽점 추출을 이용하여

얼굴인식이나 다중 객체 추적에 관한 연구 분야로 확대하여 연구하여야 한다.

참고문헌

- [1]. 이희영, 최재영, 강동구, 김홍수, 차의영, 전태수, "배경영상을 이용한 목표물 추적에 관한 연구" 멀티미디어학회 춘계학술발표논문집, 1999, pp.386-390.
- [2]. 김동호, 강은택, 김현주, 이정식, 최연성 "Fuzzy ARTMAP 신경망을 이용한 차량 번호판 인식에 관한 연구", 해양정보통신학회 춘계학술발표대회 Vol. 5, 2001, pp.625-628.
- [3]. D. Koller, J.Daniilidis and H. Nagel, "Model-based Object Tracking in Monocular Image Sequences of Road Traffic Scenes", Int'l J. of Computer Vision, Vol. 10, No. 3, pp. 257-281, Oct 1993.
- [4]. J. Yang and A. Waibel, "A Real-Time Face Tracker", IEEE Workshop on Applications of Computer Vision, pp. 142-147, 1996.
- [5]. R. C. Gonzalez and R. E. Woods, Digital Image Processing, Addison-Wesley Inc., pp. 189-200, 1995.