

가중치 정보를 이용한 영상 분할 알고리즘

김선집*, 박병준**

Image segmentation algorithm based on weight information

Sun-jib Kim*, Byung-Joon Park**

요약 영상 감시 시스템에서 핵심적인 기술은 얼마나 정확하게 검출을 할 수 있는가 하는 것이다. 객체를 정확하게 추적하기 위해서는 영상에서 배경과 객체를 정확히 분리 할 수 있어야 한다. 하지만 사람의 시각이 아닌 시스템 스스로가 배경과 객체를 정확히 구분하고, 상황을 판단하는 것은 쉽지 않다. 영상처리 시스템의 신뢰도를 높이기 위해서는 배경과 객체를 정확히 검출해 내야 객체를 정확하게 추적할 수 있게 되며 이는 시스템 제작의 성공 여부에 큰 영향을 끼친다. 본 논문은 시시각각으로 변하는 배경 환경을 정확히 파악하기 위해 배경의 변화를 관찰하여 배경과 객체를 보다 정확하게 구분할 수 있는 방안을 제안하였다.

Abstract The most important and critical to the performance of video surveillance systems is to be detected exactly how much. In order to accurately track the object must be able to accurately separate the background and object. However, the system itself rather than the human vision exactly distinguish the object and the background, to assess the situation, it is not easy. If we can accurately detect the background and the object, to be able to accurately track an object, it is possible to increase the reliability of the system, have a significant impact on the success of the entire production system. In this paper, we propose a way to distinguish more precisely the background and the object being to determine the background environment changes more accurately.

Key Words : Image Processing, Object tracking, Figure-ground segregation, Detection System, CCTV

1. 서론

컴퓨터 영상처리 분야는 CPU성능의 발달로 빠르게 발전이 이루어지면서 분야가 다양하고 세분화 되었다. 그 중 객체 추적 분야는, 컴퓨터와 카메라의 고성능화와 보급화로 그리고 감시 카메라 수요 증가에 힘입어 객체 추적에 대한 사회적 관심이 증가되고 있다[1, 2].

객체를 추적하기 위한 단계는 4단계로 나눌 수 있다.

- (1) 배경과 객체 분리
- (2) 객체 등록

- (3) 시간별 프레임 단위 사이의 움직임 추적
- (4) 추적된 데이터를 기초로 객체를 분석

영상 추적은 움직임이 발생된 시점의 영상 프레임에서 객체의 데이터 (원점, 크기, 모양 등)를 이용하여 추적을 한다. 하지만 영상에서 객체를 추적하고 검출하기 위해서는 객체 추적보다 우선 배경 화면에서 객체가 아닌 배경화면을 정확하게 인지하는 전 처리 과정을 통하여 객체가 아니면서 움직이는 픽셀을 제거해야 한다.

하나의 영상만 가지고는 각 화소에 대하여 배경과 객체가 어떠한 값을 갖게 되는지를 구분할 수 없으며 하나의 배경 영상으로 인식한다. 즉 어느

* Department of IT, Information and Communication Engineering, Hansei University

**Corresponding Author : Department of Computer Software, Seoul University(20130029@seoul.ac.kr)

Received October 03, 2016

Revised October 18, 2016

Accepted October 26, 2016

것이 배경인지 어느 것이 객체인지를 그림 한 장만 가지고는 정확히 구분할 순 없다. 이 후 새롭게 변화하는 영상의 정보를 처음 저장된 영상과 비교 분석하여 변환 데이터를 기반으로 새롭게 변환 영역을 찾아내게 된다.

객체가 없는 깨끗한 상태의 배경을 설정 하여 새롭게 변환 배경과의 차를 구하는 배경 차분 방식은 달리 물체의 움직임이 없을 경우에도 처음 기억된 배경이 변화하였으면 객체를 찾아낼 수 있다. 하지만 이 방식은 기존의 배경 영상이 완전한 배경 영상이라는 확신이 없다. 즉 주차된 자동차가 이동을 할 경우 기존에 주차에서 가려졌던 배경과 객체를 잘못 인식하는 단점이 발생할 수 있다.

본 논문에서는 배경 화면에서 객체가 아닌 노이즈를 제거 한 다음 비디오 프레임 사이에서 이전과 이후 프레임을 서로 비교하여 시간 경과에 따라 배경의 변화를 추적한다. 배경의 추적을 통하여 변화하는 영역의 움직임 영역을 찾고 이를 이용하여 상대적인 움직임 값을 추정한다. 상대적 움직임의 추정은 배경 차분법의 절대 움직임 계산에 비하여 신뢰도와 정확성이 높다.

영상의 전체 영역에서 객체로 지정된 부분이 아닌 다른 영역에서 움직임이 발생 한다면, 공간 필터를 적용하여 정교하게 움직이는 상황을 실시간으로 추정 할 수 있다. 배경의 변화를 추적 분석하는 것은 변하는 픽셀과 잡음을 구별하여 객체를 오류 없이 정확히 검출하기 위한 전처리 과정이다.

변화하는 객체를 추적하는 과정에서 객체가 아닌 배경 자체의 변화는 객체를 구분하는데 많은 혼란을 야기할 수 있다. 검출된 변화 영역의 픽셀에서 움직임이라고 추정된 부분이 있으면, 움직이는 데이터를 바탕으로 객체를 영역별로 분류한 후 배경 환경의 변화, 객체 변화 추적, 움직임 추정을 통하여 객체의 이전과 이후 변화의 형태는 전체 영상 영역에서 실시간 추적이 된다. 실험은 영상처리에 많이 이용되는 동영상상을 이용하여 실시간으로 배경 영상의 갱신과 객체의 추적이 가능함을 보여주었다.

본 논문에서는 시시각각으로 변하는 배경 환경

을 연속적으로 추적하여 배경과 객체를 분리할 수 있는 방법을 제안한다.

논문의 2장에서는 관련 연구를 살펴보고, 3장에서는 시스템 구현 방법을 설명한다. 그리고 4장에서는 실험결과를, 마지막으로 5장에서 결론을 맺는다. [그림 1]은 동영상에서 객체(사람)의 움직임을 찾은 영상이다.



[그림 1] 객체 추적 이미지
[Fig. 1] Object Tracking image

2. 관련 연구

동영상에서 객체의 움직임을 찾기 위해 수많은 기술에 대한 연구가 수행되었다. 대부분의 객체 추적 알고리즘은 각 프레임간의 특징을 비교하여 유사도를 찾는다.

본 논문은 객체와 배경을 정확하게 구분하기 위해, 연속적으로 주사되는 비디오 신호에서 이전과 이후 프레임의 정보를 비교하여 일정 프레임 사이에 변환되는 객체를 인식하고 배경의 변화를 업데이트 한다. 픽셀이 변화하는 시점인 장면전환 알고리즘은 1980년대부터 개발되기 시작하였다. 동영상에서 첫 장면과 2~3프레임의 이미지를 비교하여 움직이는 부분이 일정 임계치를 넘는지 확인하면서 다음 배경 이미지 프레임과 비교하면서 식(1)과 같이 찾는 방법이다[3].

$$x_{mp} = \left\{ \begin{aligned} & \{ (I_1(x) - I_{1-1}(x)) > T_{1(x)} \} \\ & \{ \{ I_n(x) - I_{n-1}(x) > T_n(x) \} \} \end{aligned} \right. \quad (1)$$

여기서 I는 현재의 이미지이고 n은 시간단위 프

레이름을 의미한다. 현재 이미지의 픽셀에서 이전 프레임의 이미지값을 비교한 것이 일정 임계치보다 크면 움직이는 화소로 인식한다.

이때 객체로 인식되는 부분과 배경으로 인식되는 부분을 분리한다. 움직이는 객체가 들어오면 배경 이미지 중에서 관심영역과 그렇지 않은 영역을 판단하여 식(2)와 같이 배경과 객체를 분리한다[4].

$$N = \{x : (I_n(x) - B_n(x)) > T_n(X), x \in R\} x_{mp} \quad (2)$$

$$!N = x_{nmp}$$

N은 배경 이미지에서 객체를 이루는 픽셀 값을 뺀 값이고 mp는 움직이고 있는 픽셀값, nmp는 현재 움직이지 않는 픽셀이다. 즉, 배경 이미지 픽셀에서 현재 프레임 이미지의 픽셀 값을 뺀 값이 일정 임계치보다 크면 이전 프레임에서 변한 픽셀 값이고 이 값을 움직이는 객체라 할 수 있다. 하지만 움직임을 있는 픽셀이라고 해서 전부 객체는 아니며 바람이나 조명등에 의하여 값이 바뀌는 움직임도 있다. 이를 해결하기 위하여 계속적으로 변화하는 배경 이미지에서 객체가 아닌 움직임을 정확하게 검출 하여 지속적으로 변화하는 배경 영상을 프레임을 업데이트 한다[5].

하지만 단순 비교만으로 학습을 하게 되면 두, 세 개의 프레임이나 그 이상의 프레임을 비교할 때 조명이나 다른 변화가 없음에도 [그림 2b]와 같이 바람에 움직이는 객체 이외의 흔들리는 픽셀 값이 발생하게 된다.

영상 프레임에서 시간에 따라 변화하는 배경 픽셀의 상태를 정확하게 측정 할 수 있다. 영상의 모든 픽셀은 하나의 객체를 나타내며, 원하는 객체 이외의 픽셀에는 약간의 움직임의 변화가 있다[6].

이러한 [그림2]의 움직임은 배경 이미지중 변경된 픽셀값의 레벨을 이진화 한 영상이다. 이러한 움직임은 잘못된 학습을 유도하게 되어 배경과 객체의 분리를 방해하는 요소가 된다.



[그림 2] (a) 원본이미지 (b)움직임 검출 이미지
[Fig. 2] (a)original image (b)Motion detection image

이러한 움직임은 바람이나 다른 움직임 그리고 태양이나 조명등에 의해 변하는 값으로 입력된 배경영상에서 이를 깔끔하게 제거를 하기 위해서는 여러개의 프레임 단위로 영상을 분석을 해야 한다.



[그림 3] 노이즈 제거 이미지
[Fig. 3] Removing image noise

이러한 문제의 노이즈 객체는 입력된 프레임과 다음 프레임에서 움직이는 만큼의 가중치를 누적시켜 평균값을 구하고 다음 프레임에서 입력 영상 프레임의 차이만큼 다시 계산하여 움직이는 객체 정보를 누적 및 업데이트 하여 해결하여 [그림3]과 같이 객체를 제외한 배경화면의 노이즈를 제거 할 수 있다.

3. 시스템 구현

실시간 처리 속도의 효율을 높이기 위해 RGB로 이루어진 원본 영상의 화소를 0 ~ 255레벨로 이루어진 그레이 영상으로 변환하였다. 이진 영상으로 바꾸어 프레임 비교를 하면 많은 데이터 손

실이 있으므로 그레이 영상을 이용하여 실시간 영상처리 속도 향상을 기대할 수 있고 객체를 측정을 정확하게 할 수 있다.

시각검사에서 많이 이용되는 영상차감법은 기준이 되는 검사 영상을 학습하여 획득한 영상을 기준 영상과의 논리 연산을 수행하여 기준 영상과 다른 부분을 추출해 내는 방법이며, 본 논문에서는 이전 프레임이 기준이 되면 다음에 들어올 프레임이 획득한 새로운 영상이 된다.

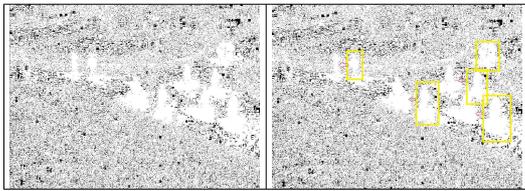


[그림 4] (a) 원본이미지 (b)그레이 변환 이미지
[Fig. 4] (a)original image (b)Gray convert images

<그림 4b>는 그레이 연산 처리를 거친 영상이다. 그레이 처리 후 건물, 보행자 및 자동차 등으로 영상이 이루어진 것을 확인 할 수 있다.

단순 영상만으로는 움직이는 객체를 찾을 수 없기에 이전 영상의 프레임과 다음 영상의 프레임을 비교하는 차감법을 이용하였다.

입력된 프레임과 다음 프레임에서 움직이는 만큼의 가중치를 누적시켜 평균값으로 영상을 구하고 다음 프레임에서 입력 영상 프레임의 차이만큼 다시 계산하여 움직이는 객체 정보를 누적 및 업데이트 한다.



[그림 5] 가중치 이진화 (b) 가중치 표시 이미지
[Fig. 5] (a) Weighted binarization (b)Weight display

<그림5a>는 이전과 이후를 15프레임 차이로 본 것이다. 움직임이 있는 픽셀은 가중치로 인하여 흰색으로 보이고 객체가 움직인 부분은 <그림5b>와 같이 표시 하였다.

가중치는 새로운 배경 화소 값 비율을 결정하는 상수 이다. α 값이 1에 가까울수록 데이터의 업데이트의 비중은 높아진다.

$$O(x, y) = I_{1Frame}(x, y) - I_{2Frame}(x, y) \quad (3)$$

$$W = (1 - \alpha) \cdot I_{MoveFrame}$$

$$O_{Move}(x, y) = I_{1MoveFrame} + I_{2MoveFrame} + \dots \rightarrow W$$

W 는 누적 가중치로 움직임이 없으면 가중치가 내려간다.

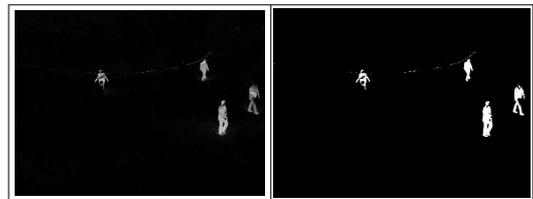
<그림 5>에서 보듯이 영상 프레임 사이에는 조그마한 움직임으로 인하여 노이즈가 많다. 이러한 노이즈를 제거하기 위하여 침식과 팽창 연산으로 관심 영역이 아닌 픽셀을 제거 하였다.

집합 A와 B에서, A의 B에 대한 침식은 $A \ominus B$ 로 표기하고 식(4)과 같이 정의 한다.

팽창은 객체 내부의 돌출부를 감소시키고 외부 돌출을 증가시켜 객체의 크기를 확대하고 배경은 축소시키는 식(5)과 같이 정의한다[7].

$$A \ominus B = w : B_w \subseteq A \quad (4)$$

$$A \oplus B = (a, b) + (u, v) : (a, b) \in A, (u, v) \in B \quad (5)$$



[그림 6] (a) 가중치 이미지 (b)침식 팽창 후 이미지
[Fig. 6] (a)Weighted images (b)After dilation erosion image

4. 실험결과

영상처리를 이용한 측정 실험은 AMD6300 CPU, Windows 7 Ultimate 32bit, Microsoft Visual Studio 2010을 이용하여 구현하였다.

영상에서 관심이 없는 영역은 전체 영상처리를 적용하지 않고 일정한 영역만 처리를 하는 것이다. 즉 일정 영역만 처리하여 알고리즘의 연산시간을 단축할 수 있다. 침식 팽창 후 영상을 보면 모든 노이즈가 제거 되지도 않았으며 객체의 픽셀 테이터도 유실된 것을 볼 수 있다. 이러한 노이즈를 제거하기 위해 가중치 값을 이용하여 관심 영역만 다시 계산 하였다.



[그림 7] 실험 이미지 1, 2, 3
[Fig. 7] Experimental Image 1, 2, 3

[표1]은 [그림 7]영상의 가중치 적용 전과 후 전체 이미지 영역에서 관심이 없는 객체(노이즈)의 남아있는 정도이다. 가중치 적용 후 관심영역 이외의 검출 영역이 거의 남아있지 않음을 볼 수 있다.

표 1. 노이즈 검출
Table 1. Noise Detection

	Weight before application	Weight after application
Experimental Image 1	3%	0.7 %
Experimental Image 2	4.5%	0.8 %
Experimental Image 3	2.8%	0.9 %

영상의 객체추적 시스템에서 복잡한 환경 문제로 배경분리가 효율이 떨어지는 문제를 해결하였다. 배경 영상에서 획득된 영상의 프레임과 다음 프레임에서 변화가 발생한 경우의 영상의 픽셀 값을 연속적으로 업데이트 하며 객체와의 분리 과정에서의 객체가 아닌 움직임을 제거하는 기법을 제안 하였다.

제안한 알고리즘을 통하여 기존의 영상 차감에

의한 객체 추적 방법에 비하여 늘어난 만큼 연산 속도는 다소 늦어 졌지만 정확한 객체 추적 및 추출이 가능함을 보였다. 객체를 추출하고 추적하는 과정에서 처리 속도는, 최대 15프레임 이상으로 기존방식에 비해 연산과정이 만든 만큼 5프레임 정도의 속도 저하가 있었다. [표2]는 실험 이미지에 따른 객체 인식 정확도를 나타낸다. [그림 8]과 같이 가중치 적용 전과 후를 계산 하였다. [표2]는 여러 객체의 인식 정확도를 계산한 결과이다.



[그림 8] (a) 가중치 이미지 (b) 가중치 적용 전 연산 (c) 가중치 적용 후 연산
[Fig. 8] (a) Weighted images (b) before Weighted process (c) after Weighted process

성능 평가에 사용된 영상은 768 x 576해상도의 24 bit 컬러영상을 24Frame/초 이다.

표 2. 객체 인식 정확도
Table 2. Object recognition accuracy

	Weight before application	Weight after application
Experimental Image 1	65%	96%
Experimental Image 2	75%	97%
Experimental Image 3	63%	98%

5. 결론

동영상에서 객체를 추적하기 전 단계로 전경을 분리하기 위해 배경 영상을 추출하여 각 프레임에 따른 배경 화면을 업데이트 하며 분석하는 방법은 보다 정확하게 객체를 추적할 수 있게 도와주는 객체 추적의 전처리 단계로 객체의 이동 경로를 추적하여 감시 시스템의 정확도를 높이는데 효과적이다.

본 논문에서는 단순 차감법을 이용하여 배경 업데이트를 하지 않는 픽셀 기반의 업데이트가 이루어질 경우의 문제를 해결하기 위해 움직임 가중치

를 이용하여 객체의 배경을 업데이트 하였다. 그 결과 객체를 프레임 단위로 비교하여 객체 검출의 정확도를 향상시켰다. 각 정보를 배경의 상태에 따라 프레임 간 정보를 융합하여 연산량의 측면에서 효율적이고, 배경의 환경이 변할 시 보다 정확하게 관심영역의 객체와 관심이 없는 객체(노이즈)를 정확히 구별할 수 있도록 하였다. 계층적 객체 검출 및 업데이트와 프레임간 분석을 통하여 객체 추적을 하기 위한 전처리 과정에서 기존 영상차감법의 문제점을 해결 하였다.

객체를 추적을 연속적으로 하면서 정보를 업데이트 하는 과정에서 연산량이 많아지는 단점을 가지고 있다. 하드웨어 속도의 향상은 멀티프로세싱과 CUDA 연산이나 최적화를 통해 문제는 개선되었지만 업데이트 시간과 타이밍의 문제는 지속적인 연구가 필요하다.

REFERENCES

[1] D. Comaniciu, V. Ramesh, and P. Meer, "Real-Time Tracking of Non-Rigid Objects Using Mean Shift," Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. II, pp. 142-149, 2000.

[2] C. Y. Jung and J. W. Han, "Intelligent image analysis event detection technology trend," Electron. Telecommun. Trends, vol. 27, no. 4, pp. 114-122, Aug. 2012.

[3] M. Cetin, W. C. Karl, and A. S. Willsky, "Edge-preserving image reconstruction for coherent imaging applications," in Proc. 2002 Int. Conf. Image Process., vol. 2, pp. 481-484, Rochester, U.S.A., Sep. 2002

[4] H. J. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," ACM Multimedia Systems, 1: pp.10-28, 1993. [6] H.B.Lu Y.J.Zhang, Y.R.Yao, "Robust Gradual Scene Change Detection," ICIP 99. Proceedings, 3 : 304-308, 1999.

[5] J. Meng, Y. Juan, S. F. Chang, "Scene change detection in a MPEG compressed video sequence," IS&T/SPIE Symposium, Proceedings, vol. 2419, Feb. 1995.

[6] H.B.Lu Y.J.Zhang, Y.R.Yao, "Robust Gradual Scene Change Detection," ICIP 99. Proceedings, 3 : 304-308, 1999

[7] Rafael C. Gnalez and Richard E. Woods, Digital Image Processing, England, pp. 40-120, EG: Addison Wesley Longman Limited, 1992.

저자약력

김 선 집 (Sun-Jib Kim)

[정회원]



- 1999년 2월 강남대학교 전자계산학과 공학사
- 2001년 2월 숭실대학교 컴퓨터학과 공학석사
- 2010년 2월 한세대학교 IT학과 공학박사
- 2012년 9월 ~ 현재 : 한세대학교 IT학부 정보통신공학과 조교수

<관심분야> 네트워크, IoT, 정보보안, CCTV, 영상처리

박 병 준 (Byung-Joon Park)

[중심회원]



- 2000년 2월 : 고려대학교 컴퓨터학부 (공학사)
- 2002년 8월 : 고려대학교 정보기기학과 (공학석사)
- 2010년 2월: 국민대학교 전산학과 (이학박사)
- 2013년 3월~현재 : 서일대학교 컴퓨터소프트웨어과 조교수

<관심분야> 영상처리, 패턴인식, 인공지능, 객체추적