

움직임 영역간 블록 정합을 이용한 반복적인 움직임 검출

고봉수, 김장형

제주대학교

The Recursive Motion Detection Using Block Matching Between Moving Regions

Bong-soo Ko, Jang-hung Kim

Cheju University

E-mail : bong9431@cheju.ac.kr, Janghhk@cheju.ac.kr

요 약

본 논문에서는 반복적인 움직임의 있는 경우, 강건하게 해결 할 수 있는 움직임 검출 알고리즘을 제시 한다. 기존에 차 영상을 이용한 움직임 검출방법은 밝기나 잡음에는 어느 정도 강건하지만, 일정 영역에서 동작하는 물체의 반복적인 움직임에 대해서는 움직임으로 오 인식 하는 문제점을 자주 발생 시킨다. 이러한 문제점을 해결하기 위해 영상에서 반복적인 움직임은 특정 영역 상에서만 움직임의 발생된다는 특징을 이용해, 움직임의 가장 많이 발생한 영역을 움직임 영역으로 설정하고, 블록 정합(Block Matching) 시켜 계산된 평균절대오차(MAE)값을 가지고 문제를 해결하는 방법을 제시한다. 실험 결과 제안된 알고리즘은 다양한 반복적인 움직임에 대해 기존의 방법들에 비해 좋은 결과를 얻을 수 있었다.

ABSTRACT

This paper presents the motion detection algorithm that can run robustly about recursive motion. The existing motion detection algorithm that uses difference image is robustly in some degree brightness or noise, but it frequently causes false alarms to temporal clutter, at the repetitive motion within a certain area. We developed a motion detection algorithm using mean absolute error(MAE) which calculates the set of Moving regions and performs block matching. The experimental results revealed that our approach is superior to existing methodologies to handling various temporal clutter.

키워드

움직임검출, 영상감시, 블록정합, 움직임해석, 반복적인움직임

1. 서 론

최근 컴퓨터 시스템의 발달로 인해 무인 영상 감시 장치에 대한 관심이 급증하고 있는 가운데 영상 감시 시스템의 주요 목적중 하나는 입력된 영상을 가지고 움직임을 감시, 저장하는 기능이다. 이러한 감시 기능을 수행 하는데 있어 입력된 영상에서 움직임의 검출되었을 경우에만 저장 할 수 있도록 하면, 감시 시스템에 저장 공간을 줄이고 차후에 움직임의 검출된 영상을 보다 쉽게 검색 할 수 있어 보다 효율적인 감시 기능의 가능하게 된다. 이러한 이유로 무인 영상 감시 장치에서 움직임을

검출하는 여러 가지 방법들이 제시 되었다.[1]

영상 감시 시스템에서 움직임을 검출하는 방법에는 다양한 방법들이 있는데, 가장 대표적인 방법에는 프레임간 차이법과 배경 차이법이 있다.

프레임 간 차이법은 현재 입력영상과 바로 이전 영상간에 픽셀들의 차를 이용하는 방법으로, 가장 일반적으로 사용되는 프레임 간 차이법에는 차영상 기법이 있다. 차영상을 이용한 움직임 검출 방법은 현재 이미지와 이전 프레임 이미지간에 픽셀 값들을 서로 비교하여 움직임을 검출하는 방법으로서 특정 임계값 이상이면 움직임이 검출된 것으로 판단한다.

하지만 이러한 차영상 기법은 처리속도는 상당히

빠르지만, 조명이나 빛에 의한 영상의 밝기 변화나 물체의 반복적인 움직임의 발생하는 영상에서는 움직임 오인식의 자주 발생하여, 신뢰성 있는 움직임 검출이 어렵다.[2]

프레임 차이법과는 달리 배경 차이법은 현재 입력 영상과 배경이 되는 영상의 차를 구하여 움직임을 검출하는 방법으로, 인접한 두 프레임을 서로 비교, 분석하여 움직임을 검출하는 것이 아니라 현재 입력 영상에서 이전 프레임 영상들로부터 배경이 되는 영상을 추정하여 현재 영상과 배경이 되는 영상을 비교 분석하여 움직임을 검출하는 방법이다.[2-5]

이러한 배경 영상은 오래된 이전 영상의 영향을 줄이고, 최근에 입력된 영상의 영향을 추가하여 계속적으로 갱신, 수정 된다. 이처럼 배경영상을 계속적으로 갱신하여 움직임을 검출하는 방법으로는 시간적 평활법과 시간적 중간치법이 있다.

시간적 평활법은 배경영상을 만들때 이전프레임들의 화소값들을 평균하여 배경영상을 만드는 방법이다. 즉 배경 영상을 만들기 위해서, 현재 입력된 영상의 이전 프레임들의 수가 N개라고 가정하면, N개의 이전 프레임들의 화소값을 다 더하여 N으로 나누어 배경영상을 만드는 방법이다. 이 방법은 이전 프레임들의 정보를 기억하기 위해 메모리의 낭비가 심하고, 배경영상을 만들때 최근 프레임의 영향과 오래전 프레임의 영향사이에 비중이 같게 나타나는 문제점이 발생한다.

배경 차이법의 다른 한 방법인 시간적 중간치법은 임의의 화소에서 이전 프레임에 나타난 값들 중에서 빈도가 높은 값을 배경영상으로 사용하는 방법이다.

이러한 방법은 일반적으로 감시 시스템에서 움직임의 발생한 경우, 각 화소에 배경이 나타나는 빈도가 움직임의 발생시, 나타나는 빈도수 보다 훨씬 크기 때문에 움직임 검출에 효율적인 방법이 될지 모르지만, 이 역시 현재 입력 영상에서부터 이전 영상들을 메모리에 저장해 두었다가 계속적으로 배경이 되는 영상을 갱신해야 하므로 많은 메모리 낭비가 심하고 이전 영상이 많으면 많을수록 처리속도가 느려진다.[6-9]

따라서 본 논문에서는 빠른 처리속도를 가지고 메모리 낭비를 줄이며, 반복적인 물체의 움직임에 대해 새로운 움직임으로 오 인식하는 문제점을 해결 할 수 있는 움직임 검출 기법을 제시 하고자 한다.

II. 제안하는 알고리즘

본 논문에서는 입력 영상에서 들어온 첫 번째 프레임 컬러 영상을 그레이 영상으로 바꾸어 배경 이미지로 지정하고, 두 번째부터 들어온 영상을 그레이 영상으로 변화시켜 지정된 배경 이미지와의 차영상을 구하여 움직임의 발생하였는지 먼저 판단한다. 이때 움직임이 발생하면 현재영상에서 배

경영상을 뺀 차영상을 이진화 시키고 이진화된 영상을 가지고 움직임 영역을 추출한다.

그 다음 블록영역의 (x,y) 픽셀 좌표를 지정된 배경과 현재 영상에 각각 적용시켜, 배경 영상과 현재 영상에 대해 블록영역을 설정하여, 두 블록영역간, 블록정합(Block Matching)을 수행 하여 평균절대오차(Mean Absolute Error : MAE)를 계산한다.[10]

이때 계산된 MAE값이 임계값보다 작으면 반복적인 물체의 움직임으로 간주하여 움직임의 발생하지 않은 것으로 판단하고, 크면 새로운 움직임으로 판단한다. 또한 MAE값이 임계값보다 크면, 배경영상을 현재 입력된 영상으로 갱신하여 저장해 두고, 만약 작으면 갱신되지 않도록 하였다.

그림 1은 본 논문에서 제안한 알고리즘의 전체 구성도이다.

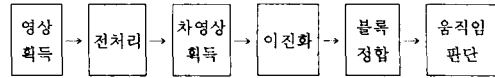


그림 1 제안한 알고리즘의 전체 구성도

본 논문에서 영상획득은 24bit RGB 컬러영상을 입력받아 256 그레이 영상으로 변환하는 전처리 과정을 거친다. 이때 256 그레이 영상으로 변환 과정은 식(1)을 사용하여 입력받은 컬러영상을 256 그레이 영상으로 변환시켜 배경영상을 얻는다.

$$g(x, y) = r(x, y) + g(x, y) + b(x, y) / 3 \quad (1)$$

식(1)에서 $g(x, y)$ 는 배경영상을 의미하고 $r(x, y)$, $g(x, y)$, $b(x, y)$ 는 각각 인접화소의 R,G,B값을 나타낸다. 이렇게 얻어진 그레이 영상은 현재영상 ($g(x, y)$)과 배경영상 ($g_b(x, y)$)간에서 식(2)를 이용하여 차영상($DI(x, y)$)를 구한다.

$$DI(x, y) = |g(x, y) - g_b(x, y)| \quad (2)$$

차영상을 획득한 후, 블록영역을 검출하기 위해 먼저 구한 차영상을 임계값과 비교하여 이진화 처리를 한다. 이때 검출된 차영상은 0인 픽셀값이 분포가 가장 많기 때문에 임계값은 식(3)을 이용하여 차영상된 영상에서 0이 아닌 모든 픽셀들을 더하여 평균값을 구하여 설정하며, 설정된 임계값과 현재 차영상의 픽셀들을 서로 비교 하여 임계값 이상이면 255로, 그렇지 않으면 0으로 할당하여 이진화 처리를 한다.

$$T = \sum_{x=0}^{x=N} \sum_{y=0}^{y=M} P(x, y) / NM$$

$$BI(x, y) = \begin{cases} 255 & (DI(x, y) > T) \\ 0 & (DI(x, y) \leq T) \end{cases} \quad (3)$$

식(3)에서 $p(x, y)$ 는 x, y 좌표에 픽셀이 값을 나타내고 NM 은 영상의 크기, $BI(x, y)$ 는 이진화된 영상, $DI(x, y)$ 는 차영상을 각각 나타내고 있다.

블록정합을 하기 위해서는 먼저 움직임 영역을 구하고 움직임 영역 좌표를 배경영상과 현재영상에 대입하여 실제 블록정합을 처리할 두 개의 움직임 영역을 획득해야 한다.

움직임 영역은 이진화된 차영상들을 행과 열로 차례대로 검색하여 255의 픽셀값을 갖는 픽셀들의 개수를 구하여, 빈도수가 많은 영역을 움직임영역으로 지정한다. 즉 255값을 갖는 픽셀들의 빈도수를 행과 열로 검색하여 임계값보다 큰 최대, 최소값을 갖는 4개의 $x_{min}, x_{max}, y_{min}, y_{max}$ 값을 구하고

$(x_{min}, y_{min})(x_{min}, y_{max})(x_{max}, y_{min})(x_{max}, y_{max})$ 를 좌표로 하는 사각형의 움직임 영역을 설정한다. 움직임 영역의 설정되면, 움직임 영역의 좌표들을 배경영상과 현재 입력 영상에 적용하여, 생성된 두영상을 블록정합 시켜 평균절대오차값을 계산한다. 반복적인 움직임인 경우에는 배경 영상과 현재입력 영상간에 유사성이 많아 평균절대오차값이 작고, 새로운 움직임의 현재영상에 들어온 경우에는 배경영상과의 유사성이 적기 때문에 평균절대오차값이 크다.

따라서 평균절대오차값이 특정 임계값보다 적으면, 반복적인 움직임으로 결정하고, 크면 새로운 움직임의 들어온 것으로 판단하여 영상을 저장한다. 식(4)는 평균절대오차값을 계산하는 수식을 보여주고 있다.

$$M(x, y) = \frac{1}{NM} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} |I_f(k+i, j+d) - B(k+x+i, l+y+j)| \quad (4)$$

여기서 NM 은 검색영역의 크기이며 I 는 현재입력영상이고, B 는 배경영상을 나타낸다.

배경이 되는 영상은 현재 입력 영상과 함께 평균절대오차값을 구하는데 기준이 되는 중요한 영상이다. 만약 반복적인 움직임의 발생한 경우 배경영상을 갱신하지 않으면, MAE값은 점점 더 커지게 되므로, 반복적인 움직임인 경우에는 배경영상을 갱신해야 할 필요가 있다. 배경영상 갱신은 식(5)와 같이 계산된 평균절대오차값, 임계값(t)보다 작으면, 현재 입력영상으로 갱신한다

$$g_{b+1}(x, y) = \begin{cases} g(x, y) & (MAE < t) \\ g_{b+0}(x, y) & (MAE \geq t) \end{cases} \quad (5)$$

위식에서 $g_{b+1}(x, y)$ 은 현재 갱신될 배경영상이고, $g_{b+0}(x, y)$ 는 이전 배경영상, 그리고 $g(x, y)$ 는 현재 입력된 영상을 각각 나타내고 있다.

III. 실험 결과

본 논문에서는 움직임의 있는 영상에서의 움직임 검출을 실험하기 위해 다음과 같은 3가지 다른 입력 영상 환경에서 PC 화상캠을 이용하여 컬러영상을 입력받아 실험하였다.

- ① 입력영상 I : 새로운 움직임의 발생한 영상
- ② 입력영상 II : 상하, 좌우로 반복 움직임영상
- ③ 입력영상 III : 불규칙적인 반복 움직임 영상

그림 2,3,4는 실험결과 각 입력영상과 배경영상에 움직임 영역을 설정한 후에 결과 영상들을 보여주고 있고, 표 1은 본 논문의 실험 결과를 차영상 기법과 비교하여 보여주고 있다. 이때 그림 2,3,4에서 (a)는 배경영상을, (b)는 현재 입력영상을 각각 나타내고 있다.

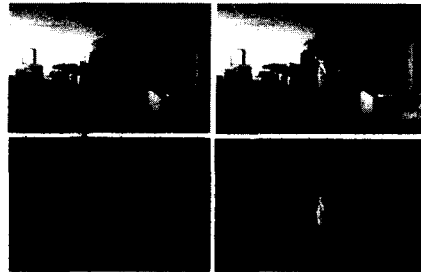


그림 2 입력영상 I

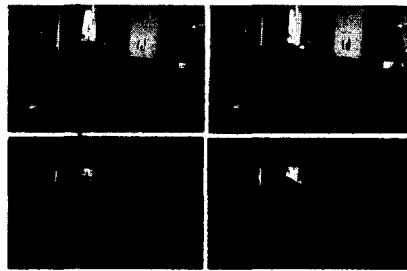


그림 3 입력영상 II

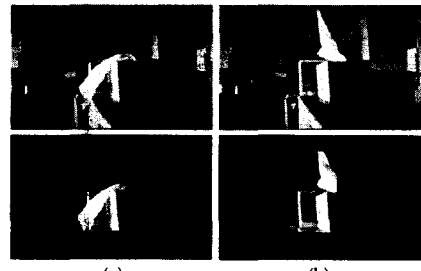


그림 4 입력영상 III

표1 움직임 검출 실험 결과

분 류	차 영상 기법			제한한 알고리즘		
	MAE	검출 오류	처리 시간	MAE	검출 오류	처리 시간
입력영상 I	3.8	6.2%	0.194 (sec)	67.6	17.3%	0.465 (sec)
입력영상 II	3.7	77.66%		13.6	10.0%	
입력영상 III	9.8	91.2%		16.5	13.3%	

움직임 검출오류를 측정 한 결과 입력영상 I 인 경우, 차 영상 기법에서는 6.2%, 제한한 알고리즘에서는 17.3%이며, 입력영상 II에서는 각각 77.66%, 10.0% 이고 입력영상 III에서는 91.2%, 13.3%로 나타났다. 처리속도는 차영상 기법이 제한한 알고리즘 보다 약 2.4배 빠르게 검출됨을 확인 할 수 있었다.

또한 입력영상 II 나 III의 환경처럼 반복적인 움직임이 있는 영상에서는 제한한 알고리즘의 차영상 기법보다 상당히 좋은 결과를 얻을 수 있었으나, 입력영상 I 처럼 새로운 움직임의 유입되는 영상인 경우에는 좋지 않은 결과를 보였다.

IV. 결론

감시 시스템에서 움직임 검출기법은 상당히 중요한 비중을 차지하는 부분으로, 얼마나 정확하게 움직임의 발생하였는지 검출하여 저장하는 것은 감시 시스템의 성능을 좌우하는 중요한 요소들 중 하나이다. 이러한 감시 시스템에서 움직임을 검출하는 방법에는 가장 빠른 처리 속도를 가지는 차영상 기법이 가장 많이 사용되고 있으나, 차 영상 기법은 밝기에 변화나 잡음 및 반복적인 물체의 움직임에 대해서는 많은 문제점이 발생한다.

본 논문에서는 이렇게 차영상의 움직임 검출기법에서 문제점으로 대두되고 있는 반복적인 움직임의 발생하였을 경우에 대해 그 해결 방법을 제시 하고 있다.

차영상을 통해 움직임의 검출되었을 때, 이 움직임의 반복적인 움직임인지, 아니면 새로운 움직임 인지를 구분하기 위해, 본 논문에서는 반복적인 움직임인 경우에 현재 입력영상과 배경영상과의 움직임의 검출된 영역에 유사성의 크고, 반면에 새로운 움직임인 경우에는 유사성의 적다는 특징을 이용하였다.

이러한 특징을 적용하기 위해 먼저 차영상을 통해 움직임의 검출되었는지 판단하여, 만약 움직임의 검출되었을 때는 차영상을 이진화 처리하여 움직임의 검출된 영역만을 움직임 영역으로 지정하도록 하였다. 이렇게 지정된 움직임 영역을 가지고 현재 입력 영상과 배경 영상과의 블록정합(Block Matching)을 시켜 평균절대오차(MAE)값을 계산하

고 이 계산된 평균절대오차(MAE)값의 크면 새로운 움직임으로, 작으면 반복적인 움직임으로 인식 할 수 있도록 하였다.

실험 결과 본 논문에서 제시한 방법으로 움직임을 검출 하였을 때 반복적인 움직임에서는 차영상 기법과 비교하여 약 77%의 성능 개선을 보였으나, 새로운 움직임의 발생한 경우에는 오히려 차영상 기법보다는 약 11%의 성능의 떨어짐을 확인 할 수 있었다. 향후 좀더 제시한 알고리즘을 개선하여 새로운 움직임 발생한 경우에 있어서, 성능을 향상시키고, 좀더 움직임의 발생한 영상에서 움직임 검출 능력을 향상시킨다면, 실제 무인감시시스템에 적용하여, 감시시스템의 성능을 향상 시킬 수 있는 결과를 얻을 수 있을 것으로 본다.

참고 문헌

- [1] 이규인,김영호,이재규,박규태 “무인감시장치 구현을 위한 단일 이동물체 추적알고리즘”, 전자공학회논문지, 제31권 B편, 제11호, Startpage 11, 1995
- [2] 하영현, 채옥삼, “차영상의 차 히스토그램을 이용한 자동 임계값 결정”, 신호 처리 종합 학술 대회 논문집 제 11권 1호, 1998
- [3] 김용훈, 이태홍, 백지흠, 조영창, 김성욱 “시간적 상관성을 이용한 움직임 추정” 한국멀티미디어 추계학술발표 논문집, Vol.2, No.2, Startpage 799, 1999
- [4] N.McFarlance, "Segmentation and Trackig of Piglets in Images,"Machine Vision Application, Vol.8,PP. 187-193,1995
- [5] 조태훈,최영규 “다중 배경 분포를 이용한 움직임 검출” 한국정보처리학회 논문집 ,Vol.8, No.4, Startpage 381, 2001
- [6] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder : Real-Time Tracking of the Human Body." IEEE Trans. Pattern Analysis and Machine Intelligence. Vol.9, No. 7, 1997
- [7] Y. Ivanov, A. Bovick, an J.Liu, "Fast Lighting Independent Background Subtraction," Technical Report No.437,MIT Media Lab. 1997
- [8] 전춘,김태식,이민걸,이주신 “동적 배경에서 Hausdorff 거리를 이용한 이동물체의 추적”, 한국통신학회 하계종합학술대회논문집, Vol .19, No.1, Startpage 537, 1999
- [9] 조영식,이주신 “부분 외곽선 정보를 이용한 이동물체의 추적 알고리즘”, 정보처리학회 논문지, Vol.8, No.5, Startpage 539, 2001
- [10] 조영창, 이태홍 “움직임 영역간 보상오차의 최소편차를 이용한 최적 블록정합 움직임추정”, 정보처리학회논문지, Vol.8, No.5, Startpage 557, 2001