

적응적 배경영상과 픽셀 간격을 이용한 움직임 검출

지 정 규* · 이 창 수* · 오 해 석

Motion Detection using Adaptive Background Image and Pixel Space

Jeong-gyu Jee* · Chang-Soo Lee* · Hae-Seok Oh**

Abstract

Security system with web camera remarkably has been developed at an Internet era. Using transmitted images from remote camera, the system can recognize current situation and take a proper action through web. Existing motion detection methods use simply difference image, background image techniques or block matching algorithm which establish initial block by set search area and find similar block. But these methods are difficult to detect exact motion because of useless noise. In this paper, the proposed method is updating changed background image as much as $N \times M$ pixel mask as time goes on after get a difference between input image and first background image.

And checking image pixel can efficiently detect motion by computing fixed distance pixel instead of operate all pixel.

Keyword : Motion Detection, Background Image, Real Time Video, Pixel Space, Edge

* 한국학술진흥재단

** 숭실대학교 컴퓨터학과

1. 서론

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 시설물감시 등에 사용되고 있다. 그런데 활용영역이 확대되면서 영상의 해상도와 전송속도, 그리고 보안시스템의 핵심인 객체영역 인식, 영상 정보의 처리, 저장, 검색 기술 등의 연구가 요구되고 있다[D. Koller, 1993 ; 이희영, 1999].

효율적인 영상 저장을 위해서는 전체 프레임을 저장하는 방법보다는 객체의 움직임을 검출하여 그 시점부터 저장하는 방법을 사용한다. 그런데 이 방법은 픽셀 변화만을 사용하기 때문에 객체의 움직임이 없어도 조명의 변화에 따라 쉽게 객체로 오인하기 쉽다.

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 움직임이 검출되는 문제점을 가지고 있다. 따라서 보다 정확한 움직임을 검출하여 객체를 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 움직임을 검출하는 방법을 제안한

다. 제안하는 방식은 기존의 움직임 검출방법을 사용할 때보다 조명이나 잡음에 강점을 가진다.

따라서 실시간 영상영역 안으로 객체유입이나 객체의 움직임을 정확히 검출할 수 있기 때문에 영상 보안시스템의 저장 공간 절약 및 객체 추적과 같은 분야에 적용이 가능하다.

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

2. 관련 연구

움직임 정보는 프레임 단위의 움직임과 각 화소에서의 움직임으로 구분된다. 프레임 움직임은 그 움직임을 추정하여 움직임 벡터를 전송한 다음 프레임의 움직임 보상을 행한다. 그러나 각 화소에서의 움직임 정보는 프레임 움직임만큼 보상된 이전 프레임과 현재 프레임과의 차이가 큰지 작은지를 나타내는 움직임 검출 정보로서 표현된다[Stephen S. Intille, 1997].

움직임을 검출하기 위한 방법에는 차영상을 이용한 방법, 블록정합기법, 배경영상방법 등이 있다.

2.1 차영상을 이용한 방법

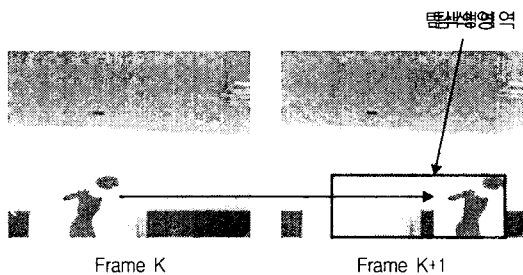
움직임이 있는 후보영역을 검출하기 위해 연속된 두 프레임간의 차영상 분석 방법을 사용한다. 차영상 분석 방법은 연속된 두 프레임간의 밝기 차이를 구한 후, 임계값을 사용하여 임계값 보다 낮은 밝기 차이를 가진 부분은 움직임이 없는 배경으로 구별하고 임계값 보다 큰 밝기 차이를 가진 부분은 움직임이 있는 물체로써

구별한다[Nikos Paragios, 2000 ; R.C Gonzalez, 1996].

이러한 임계 값의 선택은 획득한 영상 안의 잡음뿐만 아니라 시간에 따라 변하는 조명에 상당히 의존적이다. 따라서 움직임이 있는 후보영역들을 검출하기 위한 임계값은 유동적으로 선택되어야 한다. 또한 물체가 정지해 있는 상황에서는 배경으로 인식될 수 있다.

2.2 블록 정합 기법

블록 정합 기법은 현재 프레임 탐색영역 안에서 이전프레임의 지정된 블록과 가장 유사한 블록을 찾는 방법이다(그림 1). 물체가 움직이지 않다가 다시 움직이는 경우에도 추적이 가능하고 블록의 크기와 추적할 물체를 지정할 수 있다. 그러나 탐색영역 밖으로 객체가 유입되었을 경우에는 초기의 블록과 유사한 블록으로 인식하지 못하는 단점을 가지고 있다[J. Yang, 1996].

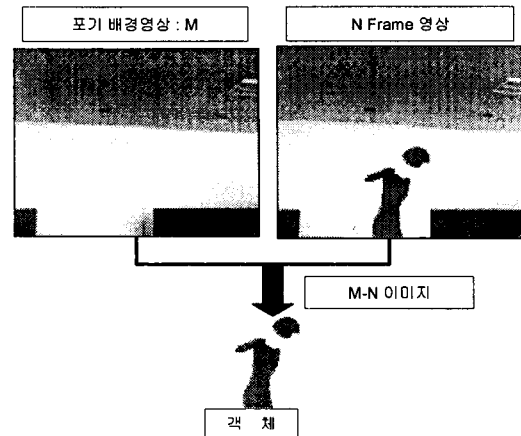


(그림 1) 탐색영역 안에서 지정된 객체 탐색

2.3 배경 영상 방법

배경 영상방법은 프레임마다 현재 프레임에서 배경 영상을 빼는 방법이다. 움직이는 물체에 대한 정확한 위치 정보와 형태 정보를 얻을 수 있다(그림 2). 그러나 기상조건, 계절의 변화, 밝기의 영향 등 조그마한 환경 변화에 많은 영향을 받기 때문에 시간의 변화에 따른 정확한

배경보상을 필요로 한다[이희영, 1999].



(그림 2) 배경영상 방법을 이용한 객체 추출

3. 실시간 움직임 검출 방법

3.1 제안 방법

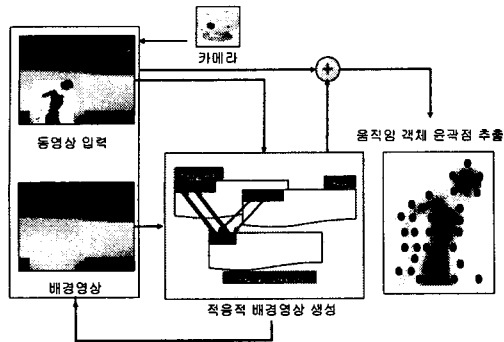
본 논문에서 제안하는 움직임 검출 방법은 카메라로부터 획득되는 초기 영상 이미지를 배경 영상 이미지로 선택한다. 배경 영상 이미지는 움직임이 없는 상태에서 획득된 순수한 배경 이미지이다.

배경영상과 카메라로부터 입력되는 영상과의 차를 이용하여 임계값이 존재하면 이미지 검사를 통하여 움직인 객체의 윤곽선에 해당하는 부분을 찾고 움직임 검출여부를 판단하게 된다.

초기의 배경영상은 시간이 지남에 따라 조명 효과나 잡음에 의해서 이미지 자체가 변화하게 된다. 따라서 초기의 배경영상만으로 입력영상과의 차를 구하게 되면 객체의 유입이나 움직임이 없어도 움직임이 있는 것처럼 인식하게 된다.

따라서 정확한 움직임 검출을 위해서 배경영상을 갱신한다.

(그림 3)은 제안하는 방법을 이용한 움직임 검출 과정을 나타내고 있다.



(그림 3) 움직임 검출 과정

본 논문에서 제안하는 움직임 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
- 단계 5. 움직임 검출 유무 판단

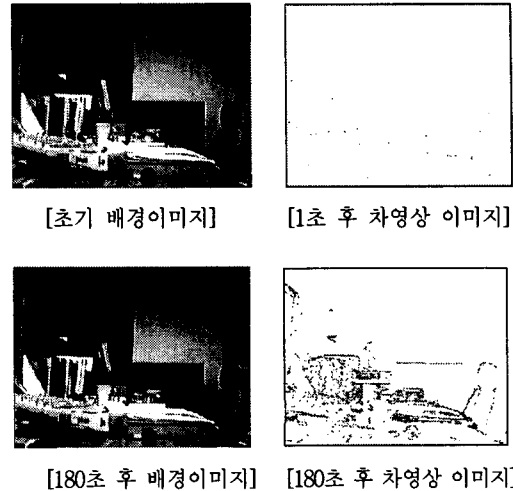
3.2 배경영상 획득

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 상황 이더라도 배경영상 자체가 변하게 된다.

변화가 없는 공간에서 조명의 변화 등과 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 발생한다.

(그림 4)는 배경영상의 시간에 따른 변화를 보여준다.



(그림 4) 배경영상의 변화

따라서 객체의 움직임이 없는 배경영상 임에도 불구하고 객체가 움직이는 것과 같은 오류가 발생 한다.

본 논문에서는 이러한 잡음을 제거하여 보다 정확한 움직임 검출을 위해서 배경영상을 갱신하는 방법을 사용한다.

전체적인 배경영상의 갱신은 많은 연산량을 필요로 하기 때문에 객체의 움직임을 검출하는데 시간이 많이 소요된다. 따라서 $N \times M$ 마스크를 사용하여 계속적으로 배경영상을 갱신하면서 객체의 움직임을 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.

$$\begin{aligned}
 R_{Channel} &= \text{abs}(R\text{Value}(\text{BackGround}_{R\text{ImageMask}}[i, j] \\
 &\quad - R\text{Value}(\text{InputImage}_{R\text{ImageMask}}[i, j]))) \\
 G_{Channel} &= \text{abs}(G\text{Value}(\text{BackGround}_{G\text{ImageMask}}[i, j] \\
 &\quad - G\text{Value}(\text{InputImage}_{G\text{ImageMask}}[i, j]))) \\
 B_{Channel} &= \text{abs}(B\text{Value}(\text{BackGround}_{B\text{ImageMask}}[i, j] \\
 &\quad - B\text{Value}(\text{InputImage}_{B\text{ImageMask}}[i, j]))) \\
 \text{if } ((R_{Channel} < \alpha) \text{ or } (G_{Channel} < \alpha) \text{ or } (B_{Channel} < \alpha)) \\
 &\quad \text{Select InputImage}[i, j] \\
 \text{else} \\
 &\quad \text{Select BackGroundImage}[i, j] \quad (1)
 \end{aligned}$$

3.3 이미지 검사 및 움직임 검출

실시간으로 배경영상에 움직이는 객체가 들어 왔는지 아닌지를 검사해야 한다. 우선적으로 배경영상과 입력영상과의 차를 이용하여 객체의 위치를 탐지한다.



(그림 5) 픽셀 간격을 이용한 객체 윤곽점 추출

(그림 5)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.

$$\begin{aligned}
 R_{Object} &= \text{abs}(R\text{Value}(\text{BackGround}_{R\text{Image}}[i, j] \\
 &\quad - R\text{Value}(\text{InputImage}_{R\text{Image}}[i, j]))) \\
 G_{Object} &= \text{abs}(G\text{Value}(\text{BackGround}_{G\text{Image}}[i, j] \\
 &\quad - G\text{Value}(\text{InputImage}_{G\text{Image}}[i, j]))) \\
 B_{Object} &= \text{abs}(B\text{Value}(\text{BackGround}_{B\text{Image}}[i, j] \\
 &\quad - B\text{Value}(\text{InputImage}_{B\text{Image}}[i, j]))) \\
 \text{if } ((R_{Object} \geq \beta) \text{ or } (G_{Object} \geq \beta) \text{ or } (B_{Object} \geq \beta)) \\
 &\quad \text{Object}[i, j] = \text{ture} \\
 \text{else} \\
 &\quad \text{Object}[i, j] = \text{false} \quad (2)
 \end{aligned}$$

그리고 R, G, B 값만으로는 명암도의 변화에도 민감한 반응을 보이기 때문에 식 (3)에 의해 H, S, I로의 변환과정을 통해 2차 검사를 하게 된다.

$$\begin{aligned}
 I &= 0.3R + 0.59G + 0.11 \\
 V_1 &= R - I = 0.7R - 0.59G - 0.11B \\
 V_2 &= B - I = -0.3R - 0.59G + 0.89B \\
 H &= \tan^{-1} \left(\frac{V_1}{V_2} \right), \quad S = \sqrt{V_1^2 + V_2^2} \quad (3)
 \end{aligned}$$

색상 변환식을 사용하여 각 채널에 대한 색조, 명도, 채도에 대한 값을 구한다[김봉기, 1998].

H는 $0 \sim 180^\circ$ 의 값이 나오는데, $B > G$ 일 경우 $H = 360^\circ - H$ 한다. S는 $0 \sim 1$ 사이의 값을 얻게 되는데, H와 S는 다시 $0 \sim 255$ 사이의 값을 갖도록 정규화 한다[이동근, 1999].

R, G, B는 약 1,600만 가지 색상으로 표현된다. H, S, I로 변환 과정은 대표색상 테이블을 이용하여 색조, 채도, 명도의 범위에 해당하면 하나의 대표색상으로 사용한다.

〈표 1〉 대표색상 매핑 테이블

대표색상	색 조	채 도	명 도
빨 강	$0 \leq H \leq 18$ $342 \leq H \leq 360$	$S \neq 0$	$0.3 \leq I \leq 0.9$
주 황	$18 \leq H \leq 54$	$S \neq 0$	$0.3 \leq I \leq 0.9$
노 랑	$54 \leq H \leq 90$	$S \neq 0$	$0.3 \leq I \leq 0.9$
연 두	$90 \leq H \leq 126$	$S \neq 0$	$0.3 \leq I \leq 0.9$
녹 색	$126 \leq H \leq 162$	$S \neq 0$	$0.3 \leq I \leq 0.9$
청 록	$162 \leq H \leq 198$	$S \neq 0$	$0.3 \leq I \leq 0.9$
파 랑	$198 \leq H \leq 234$	$S \neq 0$	$0.3 \leq I \leq 0.9$
남 색	$234 \leq H \leq 270$	$S \neq 0$	$0.3 \leq I \leq 0.9$
보 라	$270 \leq H \leq 306$	$S \neq 0$	$0.3 \leq I \leq 0.9$
자 주	$306 \leq H \leq 342$	$S \neq 0$	$0.3 \leq I \leq 0.9$
흰 색	~	$S \approx 0$	$0.85 \leq V \leq 1$
밝은 회색	~	$S \approx 0$	$0.65 \leq I \leq 0.85$
회 색	~	$S \approx 0$	$0.45 \leq I \leq 0.65$
어두운 회색	~	$S \approx 0$	$0.25 \leq I \leq 0.45$
검정색	~	~	$0 \leq I \leq 0.25$

대표색상은 <표 1>과 같이 한국 공업규격에서 정하고 있는 유채색 10가지와 무채색 5가지를 합한 15가지의 색으로 표현하였다. 여기에서 15가지 대표색상으로 표현한 것은 하나의 방법이며 다른 표현 방법을 채택할 수도 있다. 예를 들면 명도 값으로 표현하여 그 명도 값이 임계 값 범위 일 때 각 픽셀의 명도 평균 값을 사용하여 그레이 색상으로 표현할 수도 있다.

식 (3)을 적용하여 H, S, I에 대한 픽셀간 차이값을 구하여 만족하면 객체가 있는 것으로 최종 판단하게 된다.

4. 실험 결과 및 분석

4.1 실험 환경

본 실험은 명암도의 변화가 너무 크지 않은 오후시간대에 웹 카메라를 사용하여 배경영상

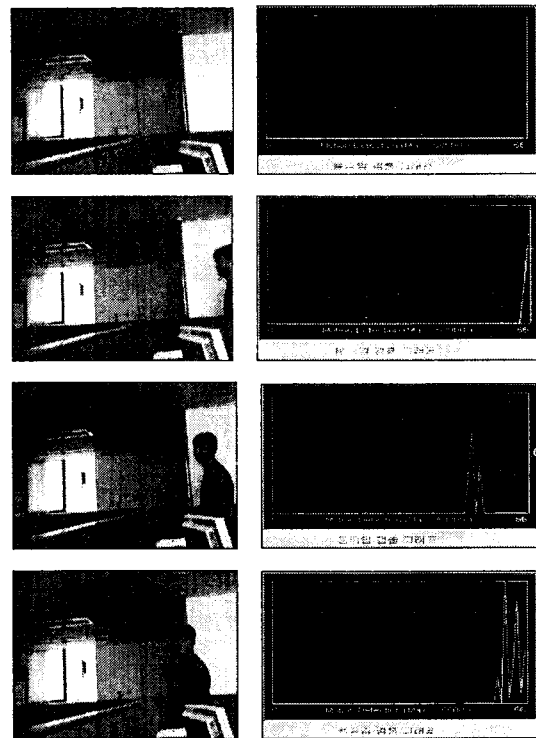
과 입력영상을 실시간으로 처리하여 실험하였다.

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

4.2 실험 결과

(그림 6)은 실시간 영상에서 움직임 객체가 영상 안으로 들어왔을 때 적응적 배경영상 변화와 픽셀간격을 이용하여 움직임을 검출하는 과정을 움직임 그래프로 표현하였다.

기존의 블록정합 방법은 객체를 블록 안에 있다는 가정 하에 움직임을 검출하기 때문에 블록 밖으로 객체가 유입되면 초기의 블록과의 유사성을 찾지 못하기 때문에 객체를 인식하지 못



(그림 6) 배경화면 변화를 이용한 움직임 검출

하는 오류가 있고, 차 영상만을 이용할 때는 잡음에 민감한 반응을 보이기 때문에 움직임 객체가 입력영상에 없다고 하더라도 움직임으로 검출하는 경우가 있었으나, 본 논문에서 제안한 적응적 배경영상을 이용하여 움직임을 검출한 결과 움직임 객체가 있을 경우만을 검색하고, 픽셀을 모든 연산에 포함시키지 않고 픽셀간격을 이용하여 객체의 윤곽점을 검색함으로써 연산시간도 단축되었다.

<표 2>는 본 논문에서 제안한 방법을 위한 임계값들과 변수에 대한 최적화 실험 결과이다. 적응적 배경영상 생성 임계값은 실험값이 25인 0.915일 때, 배경영상 이미지내의 잡음을 제거할 수 있었다.

픽셀 간격 임계값은 조밀할수록 움직임 추출시 객체 인식이 잘 이루어 졌지만, 많은 연산량을 요구하기 때문에 보다 빠른 탐색을 위해 5를 선택하여 실험하였다.

<표 2> 임계값과 변수에 대한 최적화

실험값	적응적 배경영상 생성 임계값	움직임 검출을 위한 픽셀 간격 임계값
1	0.205	0.951
2	0.151	0.903
...
5	0.210	0.851
...
25	0.915	0.595
...
30	0.887	0.496
...

<표 3>은 실험시간을 오후 1시부터 10시까지 정해서 1시부터 3시, 7시부터 10시까지는 객체의 유입이 없는 시간대로 설정하고, 4시부터 6시까지는 객체를 유입하여 움직임을 검출하는 빈도수로 실험한 결과이다.

실험은 차영상 만을 이용하여 움직임을 검출하는 방법과 배경영상을 이용하여 움직임을 검출하는 방법, 그리고 제안하는 방법인 적응적 배경영상과 픽셀간격을 이용한 것을 각각 적용하여 실시간으로 측정하였다.

<표 3> 움직임 검출 빈도수 및 검출 오류

시간 (오후)	방법 A		방법 B		방법 C	
	빈도수	오류	빈도수	오류	빈도수	오류
1	5	5	3	3	0	0
2	7	7	4	4	1	1
3	4	4	6	6	1	1
4	25	4	28	7	22	1
5	24	6	26	6	20	0
6	26	5	28	7	21	0
7	10	10	19	19	2	2
8	6	6	9	9	1	1
9	7	7	10	10	2	2
10	4	4	11	11	1	1

- 방법 A : 차영상만을 이용한 움직임 검출
- 방법 B : 배경영상을 이용한 움직임 검출
- 방법 C : 적응적 배경영상과 픽셀간격을 이용한 움직임 검출

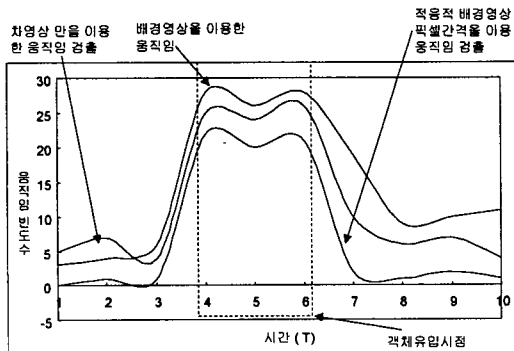
실험결과 차영상 만을 이용하여 움직임을 검출하는 방법 A는 객체의 유입이나 움직임이 없는 1시부터 3시까지의 시간대에 미세한 환경의 변화에도 민감하여 움직임이 있는 것으로 판단하고 움직임을 검출하고 있다.

배경영상을 이용하는 방법 B는 초기 배경영상을 획득하고 N 프레임의 영상의 차를 구하여 객체의 유무를 판단한다. <표 3>에서 1시와 3시 사이는 차영상 방법보다 오류검출이 적지만 시간이 지남에 따라 오류 검출횟수가 증가하고 있다.

반면에 적응적 배경영상과 픽셀간격을 이용한 움직임 검출에서는 객체가 유입된 시점인 4시부터 6시까지에 움직임을 검출하고 다른 시간대에는 조명의 변화와는 상관없이 움직임을 거

의 검출하지 않음을 알 수 있다.

(그림 7)은 움직임 검출 빈도수를 비교한 그래프이다.



(그림 7) 움직임 검출 빈도수 비교

5. 결론 및 향후 연구 방향

기존의 차 영상만을 이용한 방법은 조명에 민감한 반응을 보여 객체의 유입이나 움직임이 없었는데도 움직임을 검출하게 된다.

블록정합기법은 블록내의 객체의 움직임 측정이나, 정지하였다가 다시 움직이는 객체의 움직임 검출을 정확히 추출할 수 있다. 그러나 탐색영역 밖으로 객체가 유입되었을 경우에는 초기의 블록과 유사한 블록으로 인식하지 못하는 단점을 가지고 있다.

배경 영상방법은 프레임마다 현재 프레임에서 배경 영상을 빼는 방법이다. 움직이는 물체에 대한 정확한 위치 정보와 형태 정보를 얻을 수 있다. 그러나 배경이미지 자체는 시간에 따라 계속적으로 변하기 때문에 정확한 움직임을 검출하는데 어려움이 있다.

본 논문에서는 초기의 배경영상을 기준으로 입력영상과의 차를 구하고 시간에 따라 변화하는 배경영상을 $N \times M$ 픽셀 마스크만큼 교체하여 배경영상을 갱신해서 기존의 배경영상기법에서 배경영상 자체의 변화 때문에 생기는 문제점을 해결하였다.

이미지 픽셀 검사는 모든 픽셀을 연산에 참여시키지 않고 일정한 간격을 두고 이미지의 픽셀을 검색하여 실시간으로 처리되는 실제적인 연산량을 줄이고 정확도를 높일 수 있었다.

제안하는 적응적 배경영상 교체기법과 이미지 픽셀간격을 이용한 실시간 움직임 검출방법은 기존의 방법보다 정확한 움직임 검출방법이다.

이 방법은 영상보안시스템에서 움직이는 객체를 추적하거나, 저장 방법 개선 등에 응용할 수 있다.

향후에는 움직임 검출을 통해 객체를 추출 및 추적하는 부분과 배경영상의 획득에 대한 연구가 필요하다.

참고 문헌

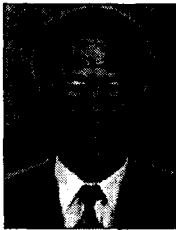
- [1] D. Koller, J. Daniilidis, H.Nagel, "Model-based Object Tracking in Monocular Image Sequences of Road Traffic Scenes," *Int'l J Computer Vision*, Vol. 10, No. 3, Oct., 1993, pp. 257-281.
- [2] Nikos Paragios and Rachid Deriche, "Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects," *IEEE Transactions on Pattern Analysis*, Vol. 22, No. 3, March, 2000.
- [3] J. Yang and A. Waibel, "A Real-Time Face Tracker," *IEEE Workshop on Applications of Computer Vision*, 1996, pp. 142-147.
- [4] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley Inc, 1995, pp. 189-200.
- [5] Stephen S. Intille, James W. Davis, Aaron F. Bobick, "Real-Time Closed-World Tracking," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June, 1997, pp. 697-703.
- [6] 이희영, 최재영, 강동구, 김홍수, 차의영, 전태

수, “배경영상을 이용한 목표물 추적에 관한 연구”, 멀티미디어학회 춘계학술발표논문집, 1999, pp. 386-390.

[7] 이동근 “색상 및 영역 특징 기반 이미지 검색 시스템”, 숭실대학교 석사학위 논문, 1999.

[8] 김봉기 “멀티미디어 데이터베이스를 위한 2 단계 내용기반 영상 검색 기법”, 숭실대학교 박사학위 논문, 1998.

■ 저자소개



지 정 규

Jeong-Gyu Jee received the B.S. degree in computer science from Seoul National University of Technology in 1987, and the M.S. and Ph.D.

degrees in computer science from Soongsil University in 1989 and 1998, respectively. From 1978 to 1996, he was employed as a project manager in the department of computing, Samho International Co. and Seoul Metropolitan Installation Management Co.. Since Feb. 1996, he has been with the Korea Research Foundation, where he is now manager in the department of information systems. His research interests are multimedia, database, and image processing.



이 창 수

Chang-Soo Lee received the B.S. degree in computer science from HanSeo University in 1999, and the M.S. degrees in computer science

from Soongsil University in 2002. He is now PH.D course in computer science from Soongsil University in 2002. His research interests are database, multimedia, and image processing.



오 해 석

Hae-Seok Oh received the B.S. degree in Applied Mathematics from Seoul National University in 1975 and the M.S and Ph.D. degrees in

computer science from Seoul National University in 1981 and 1989, respectively. He had been visiting professor of Tokyo University and Stanford University. Since 1982, he has been with the professor of Dept. of Information Science in Soongsil University. From 1997 to 1999, he was vice president of Soongsil University. His research interests are database, multimedia, and image processing.