

다중 배경 분포를 이용한 움직임 검출

조 태 훈[†] · 최 영 규^{††}

요 약

본 논문에서는 실내외의 환경에서 강건하게 동작하는 움직임 검출 알고리즘을 제안한다. 시간적 평활법(temporal mean)과 시간적 중간치법(temporal median)을 이용한 배경영상의 추출에 의한 움직임 검출 알고리즘은 일반적으로 좋은 성능을 나타내지만, 일정한 영역내에서 발생하는 "반복적인 움직임"에 대해서는 움직임 검출에서 오 인식(false alarm)을 자주 발생시킨다. 이러한 문제 해결을 위해 하나의 화소에 다중의 배경값을 허용하여, 반복적인 움직임에 대해 보다 강건한 동작을 할 수 있는 다중배경분포를 이용한 움직임 검출 알고리즘을 제안하였다. 실험 결과, 제안된 알고리즘은 다양한 반복적인 움직임에 대해서 기존의 방법들에 비해 월등한 성능을 나타내었다.

Motion Detection Using Multiple Distributions for Background

Tai-Hoon Cho[†] · Young-Kyu Choi^{††}

ABSTRACT

This paper presents a motion detection algorithm that can run robustly under both indoor and outdoor environments. Motion detection based on the background extraction using temporal mean (or median) shows good performance in normal cases, but it frequently causes false alarms to "temporal clutter", a repetitive motion within a certain area. We developed a motion detection algorithm using multiple distributions for pixels of the background that is more robust to temporal clutter. Experimental results revealed that our approach is superior to existing methodologies in handling various temporal clutter.

키워드 : 움직임 검출(motion detection), 움직임 해석(motion analysis), 영상감시(video surveillance)

1. 서 론

보안감시(security) 분야의 기존의 애널리로그 CCTV기술은 디지털 및 멀티미디어 기술의 급속한 발달로 디지털화가 급격히 이루어져, 기존에는 CCTV 카메라로 취득한 동영상을 주로 VCR에 녹화하였으나, 현재는 하드디스크 등에 디지털 동영상형태로 저장하여, 보다 선명한 화질을 유지할 수 있는 Digital Video Recorder(DVR)가 점점 보편화되고 있다. 이렇게 동영상을 디지털 형태로 저장할 때, 대부분의 감시용 CCTV에서는 평상시에 화면에서 움직임이 거의 없고, 어떤 "상황"이 돌발할 경우에만 화면에서 움직임이 발생하는 경우가 많기 때문에, 효율적으로 동영상을 저장하기 위해, 동영상에서 어떤 움직임(motion)이 포착(detection)될 경우에만 저장하는 방법이 효과적이다. 따라서, DVR에서 동영상에서 움직임 추출알고리즘은 DVR의 성능을 결정하는 중요한 요소가 되고 있는 추세다.

CCTV 카메라로부터 입력되는 동영상에서 움직임을 추출

하는 방법은 크게 프레임 차이법과 배경차이법으로 나눌 수 있다. 프레임 차이법은 모션 추출의 가장 기본적인 방법으로 인접한 프레임의 두 영상의 화소값의 차이를 이용하는 것이다. 인접한 두 영상에서 각 화소값의 차이의 절대치로 인접한 두 영상의 차영상을 만들고, 이것을 분석함으로써 모션을 추출할 수 있다. 가장 간단한 모션 검출 방법은 차영상의 모든 화소의 값을 더해 이것이 지정된 문턱치(threshold)보다 크면 모션이 검출되었다고 판단을 하는 방법이다. 이 방법은 알고리즘이 간단하다는 장점이 있으나, 잡음의 영향을 매우 많이 받으며, 영상에서 모션을 발생시키는 지역에 대한 정보를 알 수 없는 문제점이 있으며, 이 문제들을 처리하기 위해서는 후처리 과정이 필요하고 이들에 의한 처리시간이 길어진다는 단점이 있다.

배경 차이법은 현재 프레임과 기준이 되는 배경 영상의 차이를 구하는 방법이다. 즉 인접한 두 프레임을 직접 비교하는 것이 아니라 이전 프레임들로부터 배경(background)이라고 할 수 있는 영상을 유추해 내고 이 영상과 현재 프레임을 비교하여 모션을 추출하게 된다. 매번 프레임을 검사하면서 기준이 되는 배경 영상은 오래 전 프레임의

† 정 회 원 : 한국기술교육대학교 정보기술공학부 교수
 †† 종신회원 : 한국기술교육대학교 정보기술공학부 교수
 논문접수 : 2001년 7월 31일, 심사완료 : 2001년 8월 21일

영향을 줄이고 현재 프레임의 영향을 추가시키는 방법으로 특정한 방법에 따라 계속 수정된다. 이 방법에서 많이 사용되는 것으로 *시간적 평활법*과 *시간적 중간치법*으로 들 수 있다.

시간적 평활법은 기본적으로 배경 영상을 만들 때 이전 프레임들의 화소값을 평균하는 방법을 사용하는 것이다. 현재 프레임에서 임의의 화소의 배경 영상을 만들기 위해서는 현재 프레임 이전의 n 개의 프레임들의 화소값이 필요하며, 이들을 화소값을 모두 더해서 n 으로 나누면 n -프레임으로 평활화 된 현재 프레임의 배경 영상이 만들어지게 된다. 이때 n 값이 크면 보다 오래전 프레임의 영향까지 배경 영상에 포함시키는 것이고, 이 값이 작으면 최근 프레임들의 영향만을 고려하는 것이다. 그러나 이 방법은 이전 프레임의 정보를 기억하기 위한 메모리의 낭비가 많고, 또한 배경 영상을 만들 때 최근 프레임의 영향과 오래전 프레임의 영향이 같은 비중으로 나타난다는 문제점이 있다. 따라서 일반적으로 시간적 평활법을 이용할 때에는 화소값의 시간적인 평활값을 근사하는 간단한 수식을 사용하여 배경영상을 만들게 된다. 일반적으로 다음 수식과 같이 이전 배경영상과 현재 프레임의 영향을 조합하여 사용하는 방법이 많이 사용되고 있다.

$$m_n = \alpha x_n + (1 - \alpha)m_{n-1}, \quad 0 < \alpha < 1$$

이에 비하여 시간적 중간치법은 임의의 화소에서 이전 프레임에 나타난 값들 중에서 빈도가 높은 값을 배경 영상으로 사용한다는 개념이다. 예를 들어, 현재 프레임에서 임의의 화소의 배경 영상을 만들기 위해서는 현재 프레임 이전의 n 개의 프레임들의 화소값이 필요하고, 이들을 화소값을 크기순으로 정렬하여(sorting) $n/2$ 번째 크기의 화소값을 배경 영상으로 사용하는 방법이다. 이렇게 함으로써 해당 화소에 그동안 가장 빈번하게 나타났던 값이 그 화소를 대표하는 기준값이 된다는 것이다. 일반적인 감시 환경에서는 프레임의 각 화소에 배경이 나타나 있는 빈도가 모션이 나타나 있는 빈도에 비해 훨씬 크기 때문에 이러한 접근 방법이 타당성을 가질 수 있다. 그러나 이 방법 역시 정확한 미디언 값을 추출하기 위해서는 많은 이전 프레임을 저장해야 하기 때문에 이를 근사할 수 있는 수식을 사용하여 배경을 만들어 사용하게 된다[1].

전통적인 시간적 평활화 방법은 임의의 시간동안 프레임들에서 화소 단위로 평균을 취해서 배경영상을 갱신하는데, 이것의 단점은 배경 영상과 현재 프레임의 화소 명암 차이가 클 때, 평균값이 너무 빨리 변한다는 점이다. 이러한 평균값의 빠른 변화에 따라 갱신 에러의 축적 또한 커지게 된다. Fathy[2]는 배경 갱신 에러의 축적을 줄이기 위해 선택적인 배경 갱신 알고리즘을 제안하였다. Karmann[3]은 1990년에 차량의 흐름을 모니터링하기 위해 Kalman Filter

를 기반으로 하여 dynamic equation을 재귀적으로 갱신하는 방법의 배경 갱신방법을 제안하였고, Malik[4]이 이 개념을 확장하여, 하나의 배경영상이 아닌, 여러 개의 필터영상들을 갱신하였다. 그러나 이들 방법은 초기 프레임에 이동 물체가 있는 경우 정상상태가 되기 위해 시간이 너무 많이 걸린다는 문제점이 있다. 배경영상을 완전히 정적인 것으로 가정하고, 하나의 표준분포(normal distribution)로 모델링하는 방법[5, 6]은 모델을 재귀적으로 갱신함으로써, 천천히 변화하는 동영상에는 적응성이 좋으나, 배경물체의 작은 움직임을 처리할 수 없어, 옥외 영상을 모델링하는 데는 문제점이 있다. 배경차이법이 일반적으로 조명변화에 민감한데, 이에 대응하기 위한 시도로 몇 가지 방법론이 제안되었다[7-9].

위에서 설명한 기존의 대부분의 방법들이 일반적으로 좋은 성능을 나타내지만, "반복적인 움직임"(temporal clutter 또는 motion clutter)이라고 불리는 현상들을 충분히 보상할 수 없다. 예를 들어, 옥외 감시의 경우 검사지역에 많이 많은 나무가 있고 이러한 나뭇잎들이 바람에 움직이는 현상들과, 또는 실내에서의 경우 커튼이 바람에 움직이거나, 선풍기가 켜져 있는 경우 등과 같은 다양한 상황에서 반복적인 움직임이 발생한다. 이러한 현상이 발생하면 실제로 찾고자 하는 수상한 모션이 없는데도 계속 모션이 발생했다는 신호가 발생하게 된다. 일반적으로 이러한 반복적인 움직임이라고 불리는 현상들은 정확히 주기적이지는 않지만 일정한 영역에서 반복적으로 나타나게 된다. 따라서 오인식을 줄이기 위해서는 이러한 경우를 보상해 줄 수 있는 알고리즘이 필요하게 된다.

본 논문에서는 이러한 문제 해결을 위해 하나의 화소에 둘 이상의 배경값(다중 분포)을 허용하여, 반복적인 움직임에 대해 보다 강건한 동작을 할 수 있는 다중분포를 이용한 움직임 검출 알고리즘을 제안하였다. 실험결과, 제안된 알고리즘은 다양한 반복적인 움직임에 대해서 기존의 방법들에 비해 월등한 성능을 나타내었다. 본 논문의 구성은 다음과 같다. 먼저 2장에서 반복적인 움직임문제를 해결하기 위한 가장 일반적인 접근 방법인 다중 배경분포를 사용하는 방법을 설명하고 제안된 방법의 동기를 설명한다. 이러한 동기에 의해 개발된 알고리즘을 3장에서 설명하고, 다양한 환경에 대한 실험을 4장에서 기술하며, 끝으로 5장에서 결론을 맺는다.

2. 다중 배경 분포를 이용한 움직임 검출 방법의 개요 및 제안 동기

앞 장에서 설명한 시간적 평활법이나 시간적 중간치법들에서는 한 장의 배경영상을 사용하게 된다. 이것은 하나의 화소의 입장에서 볼 때, 그 이전의 프레임들에서 그 화

소를 대표하는 값으로 하나만을 만들어 사용하며, 이 값과 현재 프레임에서의 그 화소의 값의 차이가 일정 이상이면 모션이 발생한 화소라고 판단하게 되고, 그 대표값은 현재 프레임의 화소 값에 의해 개선이 되는 방식이다. 이러한 방법은 반복적인 움직임이 없는 프레임들에 대해서는 매우 효과적이지만, 반복적인 동작이 있는 프레임에서는 문제를 발생시킨다. 예를 들어, 나뭇잎이 움직이는 부분의 하나의 화소에서 생각해 볼 때 그 화소의 값은 나뭇잎으로 덮여져 있는 경우 어둡게 나타나고 반대로 잎이 움직여 그 화소를 벗어나면 밝게 빛나게 된다. 따라서 기존의 방법을 사용하면 밝은 화소값과 어두운 화소값 사이의 중간 화소값이 배경영상의 화소값이 되며, 이러한 배경값은 나뭇잎으로 덮여져 있는 경우와도 상당한 차이가 있으며, 덮여 있지 않은 경우에도 큰 차이를 나타내게 된다. 따라서 나뭇잎이 움직이는 부분에도 많은 모션이 발생했다는 신호를 보낼 수밖에 없게 된다.

이러한 문제를 해결하기 위한 가장 일반적인 방법은 배경에 대한 다중 분포를 허용하는 것이다. 즉, 앞에서 설명한 것과 같이 나뭇잎이 움직이는 경우에는, 예를 들어, 2가지의 배경값을 만들어 사용하며, 하나의 값은 나뭇잎으로 덮여져 있는 경우를 대표하는 값이 되게 하고, 다른 하나는 덮여 있지 않은 경우를 대표하는 값이 되게 한다. 이렇게 함으로써 현재 프레임의 해당 화소가 나뭇잎이 덮여져 있는 상태이거나 덮여져 있지 않은 경우에는 배경과의 차이가 매우 작아지며, 따라서 그 화소에서 모션이 나타났다는 신호가 발생하지 않게 된다.

이러한 접근 방법과 관련된 논문으로 Haritaoglu[11]는 학습기간중에 배경에 대한 통계치로부터 생성되는 이중(bi-modal) 분포를 배경의 모델로 이용하였으며, Friedman[12]은 교통감시용응에서 도로, 그림자, 차량을 모델링하기 위해 세 개의 표준분포를 사용하였다. Stauffer[10]는 이들을 일반화하여 각 화소를 임의의 개수의 Gaussian 분포의 혼합(mixture)으로 모델링하는 방법을 제안하였다. 이 방법에서는 온라인(on-line) 근사법을 사용하여 모델을 갱신하였는데, Gaussian의 값을 천천히 적용시켜서 느린 조명변화와 그림자, 반짝거림, 흔들리는 나뭇가지 등과 같은 반복적인 움직임에 의해 발생하는 다중모드분포(multi-modal distributions)에 대처하기 위해서 설계되었다. 그러나 이 알고리즘을 구현해서 다양한 환경에 대해 실험한 결과 만족스러운 결과를 얻을 수 없었는데, 알고리즘과 실험 결과를 분석해 본 결과 다음과 같은 문제점들을 발견하였다.

- 온라인 근사에서 분포의 분산(σ)이 효과적으로 계산되기가 힘들다.
- 일단 분포가 만들어지면 여러 프레임(frame)에 의해 만들어진 분포와 한두 프레임에 의해 만들어진 분포가 동일하게 취급된다.

- 다른 평균에서 발생한 분포가 프레임이 계속될수록 같은 쪽으로 움직이는 경우 두 분포를 병합할 필요가 있는데, 이에 대한 언급이 전혀 없다.

이러한 고찰의 결과로, 다음과 같은 새로운 접근 방법을 구상하게 되었다.

- 온라인 근사 방법으로는 크게 신뢰성 있는 값을 가질 수 없는 분포의 분산(σ)을 사용하지 않는다. 그 대신 각각의 분포마다 그 분포에 참여한 프레임의 수 (count)를 사용한다.
- 분포에 참여한 프레임 수를 사용함으로써 다수의 연속적인 프레임에 의해 만들어진 강건한 배경분포와 잡음 등에 의해 새롭게 만들어진 불완전한 분포의 구분이 가능하게 한다.
- 만들어진 분포를 제거하거나 병합하는 과정을 추가한다. 하나의 분포가 만들어져서 분포의 강도가 일정한 값 이하가 되면 그 분포를 제거한다. 또한 일정 강도 이상의 두 분포의 평균값이 일정한 값 이내이면 이들을 병합하여 하나의 분포로 만들어주는 과정을 추가한다. 이러한 과정은 현재의 유효한 분포가 항상 최선의 분포가 될 수 있도록 하며, 결과적으로 전체 시스템의 성능을 향상시킬 수 있다.

3. 다중 배경 분포를 이용한 움직임 검출 알고리즘

이 장에서는 앞 장에서의 고찰 결과를 바탕으로 새롭게 고안된 움직임 검출 알고리즘을 소개한다. 제안된 다중 분포를 이용한 움직임 검출 알고리즘의 전체 과정은 다음과 같다.

- 모든 화소에 대하여 다음 과정을 처리함
 1. 현재 프레임의 화소값 x 와 가장 잘 맞는 분포를 찾음
 - 1.1. 만약 잘 맞는 분포가 없으면
 - 1.1.1 현재까지의 유효분포의 수가 최대 가능한 분포 수보다 작으면 새로운 분포를 만들
 - 1.1.2 강도(weight)가 가장 낮은 분포를 하나 찾고 그 분포를 없애고 새로운 분포를 만들
 - 1.1.3 분포의 값들을 초기화
 - 1.1.4 모션인지 아닌지를 결정 함
 - 1.2. 만약 잘 맞는 분포가 있으면
 - 1.2.1 선택된 분포의 값들을 업데이트 함
 - 1.2.2 선택되지 않은 분포의 값들을 업데이트 함
 - 1.2.3 모션인지 아닌지를 결정 함
 2. 분포가 하나 이상인 경우 모든 분포를 강도 순으로 재배열 함
 3. 분포강도를 정규화 함
 4. 제거할 분포나 가까운 분포를 찾아서 처리함

3.1 자료구조(Data Structure)

하나의 화소에 대한 배경을 대표하는 값은 가우시안 분포(Gaussian Distribution)로 나타내는 것이 일반적이다. 그러나 주어진 문제에서는 연속적인 프레임에서의 모든 화소 값을 기억할 수 없으며, 따라서 Gaussian 분포를 사용하는 경우 평균과 분산을 업데이트하는 효율적인 방법이 있어야 하지만, 분산의 경우 다양한 방법을 적용한 결과 분산의 효율적인 업데이트가 이루어지지 않는다는 결론을 내렸다. 제안된 방법에서는 화소의 분포를 나타내기 위해 다음과 같은 자료구조를 사용하였다.

```
typedef struct {
    float mean ;           // 분포의 평균
    float weight ;        // 분포의 강도
    int count ;           // 분포에 포함된 프레임의 수
} PixelDist ;
```

이때 mean은 분포의 평균값(해당 화소의 배경값의 평균)을 나타내며, weight는 하나의 화소에 대해 여러 개의 배경 분포가 나타나는 경우 그 분포의 강도이며, 모든 분포의 강도의 값의 합은 1로 정규화시킨다. 이것은 하나의 화소에 대해 n개의 배경분포가 나타났다면, 그 n개의 분포에서의 강도의 값의 합은 1이 됨을 나타내는데, 강도가 높은 분포의 의미는 최근 프레임에서 자주 나타난 값들을 대표하는 분포라는 것이다. 또한 count는 연속적으로 최근프레임에서 해당 분포가 선택된 횟수를 나타내게 된다.

3.2 분포의 초기화

영상의 각각의 화소는 그 화소의 유효한 분포의 수를 나타내는 변수와 분포의 수만큼의 PixelDist 를 갖는다. 초기에는 영상내의 모든 화소에서 현재까지 유효한 분포의 수를 0으로 한다. 첫 번째 프레임이 입력되면 모든 화소는 유효한 분포를 하나씩 갖게 되는데, 그 분포의 mean은 첫 프레임에서의 화소값이 되고, weight는 1이 되며(하나의 분포만이 존재하므로), count는 1이 된다. 이러한 값들은 연속해서 다른 프레임들을 처리하게 됨에 따라 업데이트 되게 된다.

3.3 최적의 분포 선택

새로운 프레임이 입력되면 알고리즘은 모든 화소에 대하여 동일한 처리를 하게 된다. 영상의 임의의 위치에 대해 현재 프레임에서 주어진 값을 그 화소에 지금까지 유효하게 등록된 분포에 대응시켜 현재 프레임의 화소값이 잘 맞는 최적의 분포를 찾는다. 이것은 현재 프레임의 화소값과 유효 분포의 평균과의 차이가 가장 작은 분포를 먼저 찾고 이 차이가 기준값 이하인 경우 그 분포를 리턴해 준다.

프레임의 화소값과 해당 분포의 차이에 대한 기준으로 분포의 분산을 사용할 수도 있으나 여러 방법으로 실험한 결과 분산의 이용이 효율적이지 않았으며, 고정된 문턱치를

사용하는 것이 다양한 경우에 대해 더 좋은 결과를 나타냈다. 따라서 구현된 방법에서는 이러한 문턱치를 사용하여 최적의 매칭 분포를 구하게 된다.

3.4 선택된 분포가 있는 경우

현재 프레임의 화소값에 대해 유효한 배경 분포들 중에서 하나의 분포가 최적으로 선택되었다는 것은 기존에 배경으로 생각하고 있던 몇 가지의 밝기 값 중 하나와 유사한 화소값이 입력되었다는 것이다. 이 경우 먼저 선택된 분포의 평균과 강도를 새롭게 입력된 화소값을 이용하여 업데이트해야 한다. 평균과 강도의 업데이트를 위한 식은 다음과 같다.

$$m_t = (1 - \alpha_2)m_{t-1} + \alpha_2 I_t \tag{1}$$

$$w_t = (1 - \alpha_1)w_{t-1} + \alpha_1$$

$$c_t = c_{t-1} + 1$$

이때 m_{t-1} 은 이전프레임까지의 선택된 분포의 평균값이며, m_t 는 현재 프레임의 화소값(I_t)에 의해 업데이트되는 분포의 평균값이다. w_{t-1} 과 w_t 는 각각 이전 프레임까지와 현재프레임에 의해 업데이트되는 분포의 강도이다. 연속된 프레임에서 화소의 값들이 계속 같은 분포에 매핑되면 그 분포의 강도는 점점 증가해야 하고 그 이외의 다른 분포의 강도는 상대적으로 약해져야 한다. 상수 α_1 과 α_2 는 0에서 1 사이의 실수로, 배경 영상을 만들 때 이전 프레임들의 영향과 현재 프레임의 영향의 비율을 조절하는 상수이다. 즉 이 값이 작은 값이면 배경 영상을 갱신할 과거 영상에 의한 관성이 커져서 현재 영상보다는 과거의 영상에 의한 배경 효과가 강하게 나타나는 것이며, 이 값이 크면 배경 영상이 최근의 프레임들에 영향을 많이 받게 되어 배경영상의 변화가 많아지게 되는 효과가 있다. 본 논문에서는 평균의 갱신 비율과 강도의 갱신비율을 다르게 하여 사용하였으며, 보통 0.05~0.2정도의 값을 사용하였다. c_t 와 c_{t-1} 은 각각 그 분포가 최적으로 선택된 수를 기억하게 되며, 선택된 분포에 대해서는 1 증가된 값을 갖게 된다.

이렇게 선택된 분포에 대한 값들을 업데이트하고 나면, 선택되지 않은 그 이외의 분포에 대한 업데이트가 이루어진다. 이들 분포들에 대해서는 평균은 업데이트 할 필요가 없으며, 강도만을 업데이트하게 되는데, 그 식은 다음과 같다.

$$w_t = (1 - \alpha_1)w_{t-1} \tag{2}$$

이 식은 선택되지 않은 분포들은 강도가 감쇄됨을 나타내며, 감쇄의 정도는 α_1 에 의해 결정되게 된다.

이와 같이 분포에서 필요한 파라미터(배경을 나타내는 분포들)를 업데이트 한 후에는 해당 화소에서 모션이 발생했는지를 판단하여야 한다. 이 판단에는 프레임의 수(count)가

사용되는데, 최적으로 선택된 분포의 프레임 수가 기준 이
 하이면 새롭게 생겨나는 분포이며, 현재 프레임의 화소값이
 새롭게 나타나는 분포에 포함되는 경우는 모션이라고 판단
 하게 된다. 실험에서는 이 문턱치를 3으로 사용하였는데,
 이것은 인접한 프레임에서 3번 이상 나타나야지 안정된 배
 경으로 인정한다는 의미이다.

3.5 선택된 분포가 없는 경우

현재 프레임에서의 화소값과 잘 매칭되는 분포가 없는
 경우는 새로운 값이 나타났다는 것이며, 물론 모션이 발생
 한 것이다. 이러한 새로운 값은 새로운 배경 분포를 만들게
 되는데, 현재까지의 유효한 배경분포의 개수가 최대 이하이
 면 기존의 모든 분포를 남겨두고 새로운 분포를 만들어 추
 가하면 되지만, 그렇지 않은 경우 기존의 배경 분포들 중에
 서 하나를 선택하여 제거하고 새로운 배경분포를 만들어야
 한다. 기존의 배경분포에서 제거할 배경분포를 선택하는데
 는 분포의 강도를 이용한다. 즉 강도가 가장 작은 분포는
 이전의 프레임들에서 가장 적게 나타난 분포이므로 이것을
 제거하는 것이다.

새롭게 만들어지는 분포의 값은 다음과 같이 초기화된다.

$$\begin{aligned}
 m_i &= I_i & (3) \\
 w_i &= \text{MinOmega} \\
 c_i &= 1
 \end{aligned}$$

즉 평균값은 현재프레임의 화소의 밝기가 되며, 강도는 최
 소 강도(MinOmega)로 설정되고, count는 1이 된다. 이러한
 분포는 연속되는 프레임에서의 화소 값에 의해 더 큰 강도의
 분포로 살아남든지(실제의 배경 분포의 경우), 아니면 강도가
 약해져서 제거되게 된다(잡음에 의해 발생된 분포의 경우).

3.6 분포 강도의 정규화 및 분포의 무효화(제거)

임의의 화소에 대해 최적의 분포를 찾고 최적의 분포가
 있는 경우(3.4절)와 없는 경우(3.5절)를 처리하고 나면 전체
 분포의 강도는 정규화되어 있지 않으며, 따라서 이들 강도
 를 정규화하고 분포들 중에서 제거할 분포를 찾아 처리해
 주는 단계가 이어진다.

강도의 정규화는 모든 분포의 강도를 합하고 이 값으로
 각 분포의 강도값을 나누어주면 된다. 이 과정을 거치면 모
 든 분포의 강도의 합은 1이 된다. 그 다음으로 분포의 수를
 줄이는 과정이 이어진다. 분포의 제거는 다음의 두 경우에
 발생하게 된다.

먼저 분포의 강도가 작은 경우로 이것은 분포가 만들어
 지고 난 후 비슷한 값이 몇 번 발생하지 않았다(최적의 분
 포로 선택되지 못했다)는 의미이며, 이러한 경우 연속적으
 로 선택된 분포의 강도가 올라가는 만큼 선택되지 않은 분
 포의 강도가 떨어져서 발생하게 된다. 실제로 이러한 경우

는 잡음 등에 의해 일시적으로 발생한 분포로 걱정할 시간
 이 지나면 제거되게 된다.

두 번째의 경우는 충분히 강도가 큰 2개의 분포의 평균
 이 점점 같은 쪽으로 이동하여 발생하게 되는데, 처음에는
 비교적 멀리 떨어진 분포로 시작하였으나 연속적으로 프레
 임들이 입력될수록 평균이 접근하여 어느 정도 지나면 같
 은 분포로 생각해도 좋을 만큼 근사한 평균을 갖게 된다.
 이 경우는 두 분포의 평균과 강도를 이용하여 새로운 평균
 을 만들고 강도 값은 두 강도의 값을 합하게 된다. 이 과정
 은 유사한 분포를 계속적으로 유효화 하여 새로운 분포가
 발생되는 것을 막는 문제를 제거하게 된다. 실제로 실험에
 서 구현해 본 결과 상당히 좋은 성능을 나타내었다.

4. 실험 및 고찰

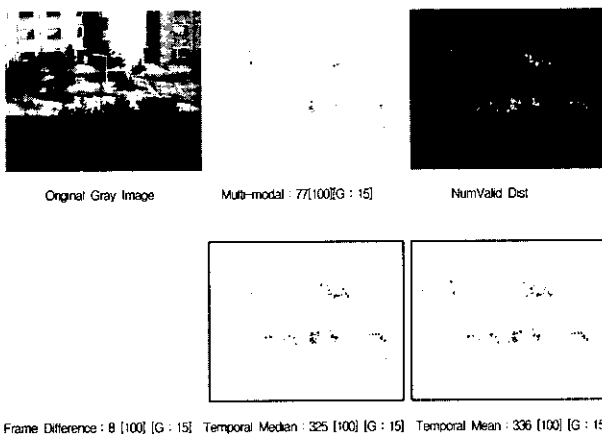
제안된 알고리즘은 기존의 알고리즘들에서 처리하기 어
 려운 반복적인 움직임에 대해 우월성을 갖는 방법이다. 이
 방법의 특성을 요약하면 다음과 같다.

- 각 화소마다 배경 밝기(분포)의 개수를 하나 이상으로
 할 수 있음
- 각 화소마다 현재의 유효한 분포의 수가 각기 다르게
 하여 반복적인 움직임이 있는 영상 영역에는 다수의
 배경 분포가 존재하고, 그렇지 않은 영역에는 하나의
 배경 분포가 생김
- 최대 가능한 분포의 수를 제한하여 처리시간을 효율화
 시키고, 무제한적인 분포 생성에 의한 문제를 없앴
- 평균이나 강도와 같은 분포 특성을 효율적으로 업데이트
 하는 방법 제공
- 여러 번 나타나지 않은 분포는 강도를 낮추어 결국 유효
 하지 않게 됨
- 새롭게 나타나는 값은 새로운 분포를 만들게 됨
- 두 분포의 평균이 한쪽으로 이동하는 경우 이들을 하
 나의 분포로 통합할 수 있음
- 모든 분포의 강도를 정규화 함

이러한 특성에 의해 반복적인 움직임이 있는 영상에 대
 해 다양하게 실험한 결과 이전의 방법에 비해 우월함을 알
 수 있었다.

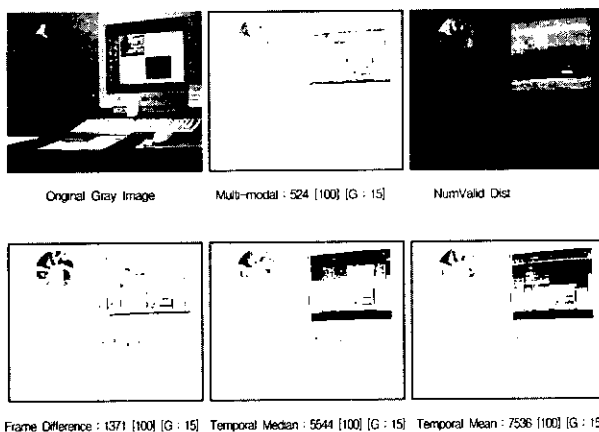
먼저 (그림 1)은 바람이 불어서 나뭇가지들이 움직이고
 있는 상황을 보여준다. 그림의 좌상단의 영상은 원 영상을
 나타내며, 좌하단에서부터 오른쪽으로 각각 프레임차이법,
 시간적 중간치법, 시간적 평활법에 의한 결과 영상을 보여
 주고 있다(검게 나타난 부분들이 모션이 발생한 부분임).
 또한 영상의 가장자리에 굵은 선이 그려져 있는 영상은 모
 션이 발생한 경우를 나타낸다. 그리고 그림의 중앙 상단의
 영상이 제안된 방법에 의해 모션을 검출한 결과를 보여주

고 있으며, 우측 상단의 영상은 제안된 방법에서 현재 프레임에서의 유효한 분포의 개수를 나타내고 있다. 실험에서는 최대 가능한 분포의 수를 3으로 제한하여 사용하였다. 그림에서 검은 부분은 배경분포가 하나인 영역이고 밝기가 더 밝은 부분은 배경분포가 2개, 그리고 가장 밝은 부분이 최대 3개의 분포가 유효하게 되어 있는 영역이다. 이 영상에서 나뭇가지가 흔들리는 부분은 배경분포의 개수가 하나 이상이며, 그렇지 않은 부분들은 모두 배경분포의 개수가 1임을 알 수 있다. 이렇게 배경분포의 개수를 다양하게 허용함으로써 나뭇가지가 흔들릴 때 발생할 수 있는 오 인식(false alarm)의 회수를 크게 줄일 수 있었다. 그림에서 하단의 기존의 방법(temporal median = 325, temporal mean = 336)들과 비교해 보면 같은 파라미터를 사용하였을 때 모션으로 판단되는 화소의 수가 절반이상(multi-modal = 77)으로 줄어드는 것을 알 수 있다.

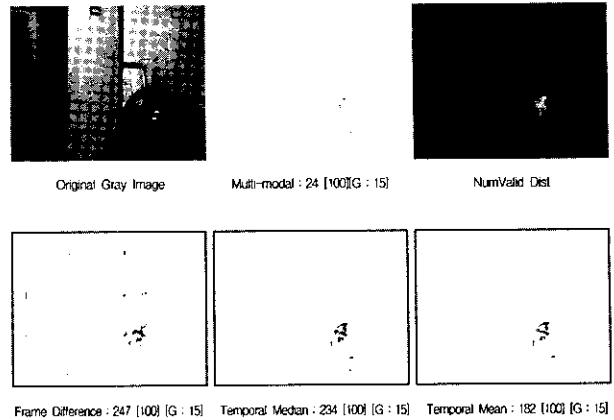


(그림 1) 바람에 의한 나무의 반복적인 동작이 있는 경우

(그림 2)와 (그림 3)은 모니터와 선풍기를 보여주고 있는 실험이다. 보통 사무실과 같은 실내에서는 선풍기나 모니터가 있는 경우가 많으며, 이들의 움직임 역시 반복적인 움직임을 나타내는 움직임이다. 이와 같은 경우에 대해서도

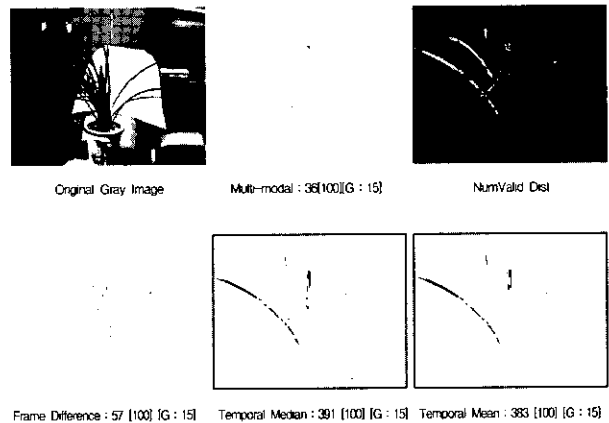


(그림 2) 모니터와 선풍기(반복적인 동작이 있는 경우)



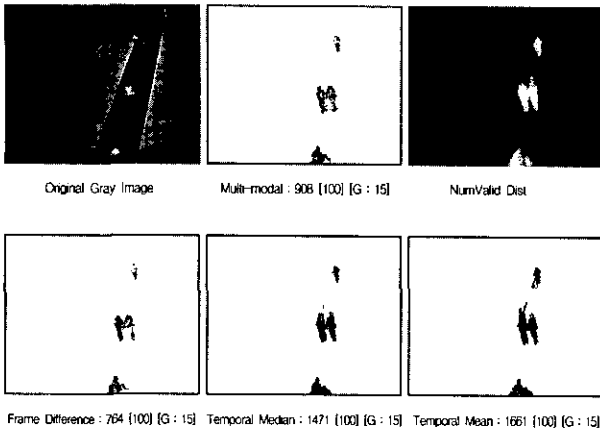
(그림 3) 선풍기(반복적인 동작이 있는 경우)

제안된 방법이 좋은 성능을 나타내고 있는데, 반복적인 움직임이 나타나는 영역에서 모션의 발생이 1/3~1/6로 줄어드는 효과를 나타내고 있다. (그림 4)는 화분을 보여주고 있으며, 이 경우에도 제안된 방법에 의한 모션 추출 결과가 탁월함을 알 수 있다.



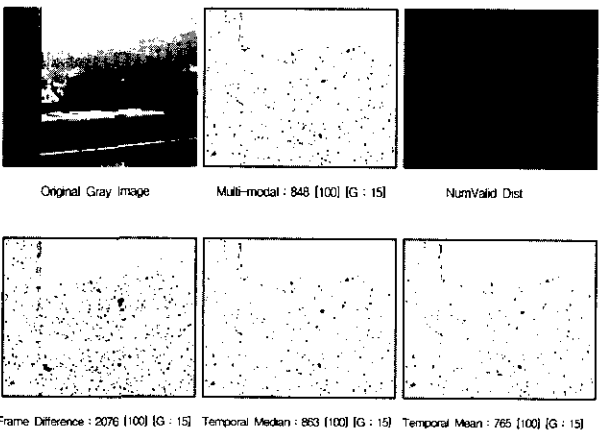
(그림 4) 흔들리는 난초(반복적인 동작이 있는 경우)

(그림 5)은 반복적인 움직임이 없는 동영상 데이터이다. 결과에서 알 수 있듯이 반복적인 움직임이 없는 일반 동영상에 대해서도 제안된 방법이 잘 동작함을 알 수 있다. 일반적으로 하나 이상의 배경분포를 허용하는 경우 모션이 일어나는 화소의 수가 현저히 줄어드는 문제가 발생할 수 있으나, 제안된 알고리즘에 의해 추출된 모션이 발생한 화소의 수는 시간적 중간치법이나 시간적 평활법의 경우에 비해 60~70% 정도로 나타났다. 그러나 시간적 평활법이나 시간적 중간치법에서는 물체가 움직이고 난 이후에 길다란 꼬리가 나올 수 있는데, 제안된 방법에서는 그러한 문제가 발생하지 않았다. 실제로 그림에서 시간적 평활법의 결과의 경우 사람이 지나가고 난 자리에서도 모션이 추출되어 꼬리가 생겼으며, 이러한 오류에 의해 모션 발생 화소의 수가 증가되었으나, 제안된 방법에서는 그러한 영향은 전혀 나타나지 않았다.



(그림 5) 반복적인 움직임이 없는 일반적인 경우

마지막으로 (그림 6)은 눈오는 날에 실험한 결과이다. 실험한 결과 눈이 내리는 것과 같은 현상은 일정한 반복적인 움직임이라고 볼 수 없을 정도로 불규칙하였다. 따라서 제안된 방법도 기존의 방법들과 같이 크게 효과적인 결과를 나타내지 못하였다. 눈이 내리는 현상은 영상에서는 매우 랜덤하게 화면에 점들이 점멸하는 현상으로 나타나며, 눈송이의 크기 또한 카메라와의 거리에 의해 매우 다양하게 나타났다. 따라서 비가 내리거나 특히 눈이 내리는 경우는 반복적인 움직임으로 생각할 수 없는 현상이라는 결론을 내렸다.



(그림 6) 눈오는 날 (랜덤한 동작)

<표 1>은 전체 실험결과를 보여주고 있는데, 제안된 방법은 다양한 반복적인 움직임에 대해서 기존의 방법들에 비해 월등한 성능을 나타냈으며, 반복적인 동작이 없는 일반적인 경우에 대해서도 기존의 방법과 대등한 결과를 나타내는 것을 알 수 있다.

제안된 방법의 문제점으로는 <표 1>에서 알 수 있듯이 각 화소에 여러 개의 분포를 허용함에 따른 계산량이 많아지는 것을 들 수 있다. 본 실험에서는 Pentium-700MHz 정도의 컴퓨터에서 320x240 해상도에 대해 처리하였으며, 하나의 화소에서 최대 3개까지의 배경분포를 만들 수 있게

허용하였다. 본 실험에서는 초당 5~10프레임을 처리하였는데, 컴퓨터의 발달에 따라 이러한 계산량의 문제는 극복될 것으로 보이며, 제안된 방법이 반복적인 움직임이 많은 분야에서 오 인식을 크게 줄일 수 있는 강건한 방법임을 알 수 있었고, 다양한 응용분야에 적용될 수 있으리라고 기대된다.

<표 1> 실험 결과표

| 입력 데이터 | 프레임 수 (320X240) | 프레임당 평균 모션 화소 수 | | | |
|---------------|--------------------|------------------|-----------------|---------------|-------------|
| | | Frame Difference | Temporal Median | Temporal Mean | Multi-Modal |
| 그림 1 : 나무 | 300 frames | 12 | 305 | 321 | 77 |
| 그림 2 : 모니터 | 300 frames | 1380 | 5014 | 6993 | 532 |
| 그림 3 : 선풍기 | 300 frames | 256 | 220 | 174 | 25 |
| 그림 4 : 난초 | 300 frames | 53 | 394 | 380 | 41 |
| 그림 5 : 교경 | 500 frames | 748 | 1529 | 1670 | 884 |
| 그림 6 : 눈 | 200 frames | 2102 | 883 | 772 | 850 |
| 프레임당 평균 처리 시간 | 단위 [msec] | 14 msec | 19 msec | 35 msec | 145 msec |

5. 결 론

본 논문에서는 실내외의 환경에서 강건하게 동작하는 움직임 검출 알고리즘을 제안하였다. 시간적 평활법과 시간적 중간치법을 이용한 배경영상의 추출에 의한 움직임 검출 알고리즘은 일반적으로 좋은 성능을 나타내지만, 일정한 영역내에서 발생하는 반복적인 움직임에 대해서는 움직임 검출에서 오 인식을 자주 발생시킨다. 본 논문에서는 이러한 문제를 해결하기 위하여 하나의 화소에 다중의 배경값을 허용하여 반복적인 움직임에 대해 보다 강건한 동작을 할 수 있는 다중배경분포를 이용한 움직임 검출 알고리즘을 제안하였다. 실험 결과, 제안된 알고리즘은 다양한 종류의 반복적인 움직임들에 대해서 기존의 방법들에 비해 월등한 성능을 나타냈으며, 반복적인 동작이 없는 일반적인 경우에 대해서도 기존의 방법과 대등한 결과를 나타내었다.

참 고 문 헌

[1] N. McFarlane and C. Schofield, "Segmentation and Tracking of Piglets in Images," *Machine Vision Application*, Vol. 8, pp.187-193, 1995.
 [2] M. Fathy and M. Y. Siyal, "An Image Detection Technique Based on Morphological Edge Detection and Background Differencing for Real-Time Traffic Analysis," *Pattern Recognition Letter*, Vol.16, pp.1321-1330, 1995.
 [3] K. P. Karmann, A. Brandt, and R. Gerl, "Moving Object Segmentation Based on Adaptive Reference Images," *Proc. Signal Processing V: Theories and Applications, (Barcelona, Spain, Sep. 18-21)*, L. Torres, E. Masgrau, M. A. La-

gunas, Eds., Elsevier, pp.951-954, 1990.

[4] J. Malik and S. Russel, "A Machine Vision Based surveillance System for California Roads," *PATH Project MOU-83 Final Report, University of California, Berkeley*, 1994.

[5] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfindex : Real-Time Tracking of the Human Body," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.9, No.7, 1997.

[6] T. Olson and F. Brill, "Moving Object Detection and Event Recognition Algorithms for Smart Cameras," *Proc. DARPA Image Understanding Workshop*, pp.159-175, 1997.

[7] Y. Ivanov, A. Bovick, and J. Liu, "Fast Lighting Independent Background Subtraction," *Technical Report No.437, MIT Media Lab.*, 1997.

[8] N. Oliver, B. Rosario, and A. Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions," *Proc. Int'l Conf. Vision Systems '99*, Jan. 1999.

[9] C. Ridder, O. Munkelt, and H. Kirchner, "Adaptive Background Estimation and Foreground Detection Using Kalman Filtering," *Proc. Int'l Conf. Recent Advances in Mechatronics, ICRAM '95*, pp.193-199, 1995.

[10] C. Stauffer and W. E. L. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," *Proc. Computer Vision and Pattern Recognition (CVPR) 99*, Fort Collins, Colorado, June, 1999.

[11] I. Haritaoglu, D. Harwood, and L. S. Davis, "W⁴ : Real-Time Surveillance of People and Their Activities," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.22, No.8, pp.809-830, 2000.

[12] N. Friedman and S. Russell, "Image Segmentation in Video

Sequences : A Probabilistic Approach," *Uncertainty in Artificial Intelligence*, 1997.



조 태 훈

e-mail : thcho@kut.ac.kr
1981년 서울대학교 전자공학과 졸업(학사)
1983년 한국과학기술원 전기및전자공학과 (석사)
1991년 Virginia Polytechnic Institute & State University(박사)

1983년~1986년 국방부, 연구원
1991년~1992년 VPI & SU, Research Scientist
1992년~1998년 LG산전(주) 연구소, 책임/수석 연구원
1998년~현재 한국기술교육대학교 정보기술공학부 조교수
관심분야 : Machine Vision, 영상처리/분석, 패턴인식, Neural Networks



최 영 규

e-mail : ykchoi@kut.ac.kr
1989년 경북대학교 전자공학과 졸업(학사)
1991년 한국과학기술원 전기및전자공학과 (석사)
1995년 한국과학기술원 전기및전자공학과 (박사)

1995년~1999년 LG산전연구소 영상연구실 선임연구원
1999년~현재 한국기술교육대학교 정보기술공학부 조교수
관심분야 : Computer Vision, Volume Visualization, 가상현실, 멀티미디어 등