

벡터 미디언을 이용한 비디오 영상의 온라인 배경 추출

김 준 철[†] · 박 은 종^{**} · 이 준 환^{***}

요 약

배경추출은 비디오 감시 시스템에서 움직이는 물체를 찾는 데 중요한 기술이다. 본 논문에서는 벡터 정렬을 이용한 새로운 온라인 컬러 배경 추출 방법을 제안한다. 제안된 방법에서 배경은 물체보다 발생빈도가 높다는 사실을 이용하여, 연속된 프레임의 컬러화소 값들의 벡터 미디언을 그 화소에서의 배경이라 간주한다. 본 알고리즘에서 현재 프레임의 물체는 얻어진 배경과의 거리가 문턱치보다 큰 화소들의 집합으로 구성된다. 알고리즘의 성능을 평가하기 위하여 온라인 가우시안 혼합 모델(Gaussian Mixture Model)을 이용한 다중 배경추출 방법과 비교하였으며, 비교결과 유사 또는 우월한 실험 결과를 확인하였다.

키워드 : 배경추출, 비디오감시시스템, 벡터미디언, 온라인알고리즘

On-line Background Extraction in Video Image Using Vector Median

Joonchoel Kim[†] · Eunjong Park^{**} · Joonwhoan Lee^{***}

ABSTRACT

Background extraction is an important technique to find the moving objects in video surveillance system. This paper proposes a new on-line background extraction method for color video using vector order statistics. In the proposed method, using the fact that background occurs more frequently than objects, the vector median of color pixels in consecutive frames is treated as background at the position. Also, the objects of current frame are consisted of the set of pixels whose distance from background pixel is larger than threshold. In the paper, the proposed method is compared with the on line multiple background extraction based on Gaussian mixture model(GMM) in order to evaluate the performance. As the result, its performance is similar or superior to the method based on GMM.

Key Words : background extraction, video surveillance system, vector median, on-line algorithm

1. 서 론

비디오를 이용한 감시 시스템은 연속된 영상에서 움직이는 물체를 찾아 분할하고, 그 움직임을 예측 추적하여, 필요한 해석을 제공한다. 이러한 감시시스템의 핵심은 추적대상의 물체를 검출하는 것으로 통상 물체 검출에는 영상차이(image difference) 방법을 사용한다. 영상차이 방법에는 비디오의 전후 프레임의 차이를 이용하는 프레임 차이법과 시간에 따라 느리게 변화하는 배경영상을 생성하여 연속되는 프레임과의 차이를 이용하는 배경차이법이 사용된다.[1-2]

프레임 차이법은 이전 프레임과 현재 프레임과의 차이를 비교하여 그 차가 큰 부분을 움직이는 물체로 간주하는 방

법으로 비교적 적은 메모리가 소요되고, 특별히 배경을 추출할 필요가 없기 때문에 계산량이 적다. 그러나, 느리게 변화하는 물체의 경우에는 움직이는 물체를 추출할 수 없으며, 빠르게 변화하는 물체라도 차 영상에는 물체의 움직임 전후의 차에 해당하는 면적만이 남기 때문에 물체 전체를 찾아내기 위해서는 통상 다양한 후처리가 필요하다.

배경 차이법은 고정된 카메라로부터 얻은 일련의 비디오 영상에서 변화가 없는 배경 영상을 획득한 후, 그 배경 영상과 현저히 다른 화소들을 움직이는 물체로 간주하여 분할하는 방법으로, 정확한 물체 추출을 위해서는 강인하고 적응적인 배경영상의 추출은 대단히 중요하다. 이를 위해서 온라인 배경영상이 갱신되어야 하며, 시간에 따른 조명의 변화를 고려하며 배경을 추출할 수 있어야 한다. 우수한 배경 추출 알고리즘을 판별하는 평가적도는 조명 및 날씨 변화와 반복적으로 움직이는 물체들(예-바람에 흔들리는 나뭇잎) 그리고 느리게 움직이는 물체에 어떻게 대응하는가와 배경을 추출하는데 걸리는 계산량 및 소요되는 메모리 등이다.[3]

* 본 연구는 산업자원부의 지역혁신 인력양성 사업에 의해 부분적으로 지원되었음.

† 정 회 원 : 서남대학교 집적전자공학과 조교수

** 정 회 원 : 전북대학교 영상공학과 박사과정

*** 정 회 원 : 전북대학교 전자정보공학부 교수

논문접수 : 2006년 4월 11일, 심사완료 : 2006년 8월 21일

통상 온라인 배경추출 방법은 두 가지 원리를 가정한다. 그 하나는 배경영상의 화소는 동영상 프레임의 시간축 상에 늘어 놓았을때 그 화소 위치에서 가장 많은 빈도를 갖는다는 가정하에 출발하여 현재 프레임을 포함하여 대표치를 추정하는 방법이며[4-9], 다른 하나는 각 화소 위치에서 물체와 그림자, 배경을 포함하는 다중모드의 분포를 갖는다는 가정을 바탕으로 한 방법이다.[10-12]

비교적 간단한 전자의 방법에는 현재 프레임을 포함하여 이전의 수개의 프레임의 평균치로 배경을 가정하는 시간적 평활법과 현재 프레임을 포함하여 이전의 수개의 프레임의 미디언을 배경으로 가정하는 시간적 중간치법등이 있다. 시간적 평활법은 배경영상을 만들 때 이전 프레임들의 화소값을 평균(moving average)하여 사용하며, 시간평활에 기여하는 이전 프레임의 수가 많으면 오래된 프레임의 영향까지 배경에 포함시키는 것이며, 이 값이 작으면 최근 프레임의 영향만 고려하여 배경 영상을 추출한다. 이 방법은 평균치를 대표치로 추정할 경우에 발생하는 문제를 그대로 담고 있다. 즉 평균치는 특이데이터(outlier)가 존재하는 경우에는 부적절한 대표치이기 때문에, 한 픽셀 위치가 배경들로만 이루어진 경우에는 적합할 수 있어도 물체 또는 그림자가 점유하는 경우에는 이들이 특이데이터로 작용하여 신뢰성있는 배경추출이 어렵게 된다.

시간적 중간치법은 임의의 화소에서 현재 프레임을 포함한 이전 프레임에 나타난 값들을 크기 순서로 정렬하여 미디언 크기의 값을 배경영상으로 사용하는 방법이다. 이 방법은 시간적 평균치 방법에 비해 특이데이터에 민감하지 않으며, 배경부분이 물체나 그림자 부분보다 더 많은 빈도를 가지면 신뢰성있는 배경추출이 가능하다. 그러나 컬러영상과 같이 벡터의 경우에는 미디언을 새롭게 정의할 필요가 있으며, 순서정렬 때문에 계산량은 시간적 평균치 방법에 비해 많아지게 된다. 이를 줄이기 위해 미디언 계산에 소요되는 이전 프레임의 수를 제약한다면 물체 또는 그림자의 빈도가 많은 시간대에서는 배경영상에 물체 또는 그림자 부분이 포함되어 배경으로 적합하지 못한 경우가 발생한다.

한편 한 화소 위치에서 연속되는 프레임의 화소 값들을 배경, 물체, 그림자 등의 다중분포(multiple distribution)으로 가정하는 경우, 통상 다중 가우시안 분포로 가정하고 기대치 최대화(expectation maximization)에 의해 온라인으로 각각의 분포의 평균과 분산 등의 계수를 추정하여 배경을 생성하거나 물체와 그림자를 분별해 낸다. 이 방법의 경우에는 안정된 계수를 찾기 위해 장시간의 화소값을 저장하여야 되며 이는 충분히 시간이 지난 뒤 신뢰성있는 배경을 추출할 수 있다는 의미이며, 빠르게 변화하는 조명환경이나 나뭇잎의 흔들림과 같이 텍스처가 혼잡한 영역에서는 배경과 물체 검출시 오검출을 야기할 수 있다. Stauffer[10]는 이 방법에 망각 요인(forgetting factor)을 도입하여 현장 상황에 대한 적응성을 높였으며, 최근 Lee[12]는 매 프레임마다 가우시안 계수를 갱신하는데 있어서 환경의 변화에 빠르게 적응하는 새로운 보류 요인(retention factor)을 도입하여 계수

수렴 속도를 개선하였다.

본 논문에서는 기존의 시간적 중간치법을 보완하여 컬러 영상에서도 적용할 수 있는 벡터 정렬을 이용한 배경 추출 방법을 제안하였다. 또한 미디언을 배경으로 하는 경우의 단점인 시간축의 프레임을 샘플링하는 방법으로 계산량 문제를 개선하면서 안정된 배경과 물체를 구별하는 알고리즘을 제안한다. 또한 제안된 알고리즘을 평가하기 위해 현재까지 제안된 배경 추출 방법 중 가장 우수한 성능을 보인 Lee의 가우시안 혼합 모델을 이용한 다중 배경 분포 방법과 비교하였다. 실험 결과 제안된 방법은 가우시안 혼합모델을 이용한 방법과 유사 또는 우월한 결과를 확인하였다.

본 논문의 구성은 다음과 같다. 2장에서는 가우시안 혼합 모델을 기술하였고, 3장에서는 제안된 벡터 정렬법을 이용한 배경 및 움직이는 물체를 추출하는 방법을 기술하였으며 4장에서는 실험 및 고찰을 하였고 끝으로 5장에서는 결론을 맺었다.

2. 가우시안 혼합 모델(Gaussian Mixture Model : 이하 GMM)

GMM[10]은 각 화소를 임의의 개수의 가우시안 분포의 혼합으로 모델링하는 방법이다. 이 방법에서는 온라인 근사법을 이용하여 모델의 계수를 갱신하는데, 계수의 값을 조절하여 다양한 환경에 적응하는 배경을 추출 할 수 있다.

시간 t 에서 화소 (x_0, y_0) 의 히스토리는 다음과 같다.

$$\{X_1, \dots, X_t\} = \{(x_0, y_0) : 1 \leq t \leq t\} \quad (1)$$

여기서 I 는 영상의 시퀀스(sequence)이다. 각 화소의 최근 히스토리를 K 개의 GMM으로 나타내면 식(2)와 같다.

$$P(X_t) = \sum_{j=1}^K w_{j,t} * \eta(X_t, \mu_{j,t}, \Sigma_{j,t}) \quad (2)$$

식 (2)에서 $w_{j,t}$ 는 시간 t 에서 j 번째 가우시안 모델의 가중 계수이다. 또한 하나의 화소에 여러 개의 배경 분포가 나타나는 경우 그 분포의 강도이며, 모든 분포의 강도의 합은 1로 정규화시킨다. 강도가 높은 분포의 의미는 최근 프레임에 자주 나타나는 값들을 대표하는 분포이다. $\mu_{j,t}$ 와 $\Sigma_{j,t}$ 는 시간 t 에서 j 번째 가우시안의 평균과 분산행렬이다. η 는 가우시안 확률 분포 함수이다.

$$\eta(X, \mu, \Sigma) = \frac{1}{(2\pi)^2 |\Sigma|^2} e^{-\frac{1}{2}(X - \mu)^T \Sigma^{-1} (X - \mu)} \quad (3)$$

예를 들어 화소의 분포를 그림자, 움직이는 물체, 배경으로 가정할 경우 K 의 값은 3이다. 또한 계산상의 이유로 분

산 행렬을 $\Sigma_{k,i} = \sigma_k^2 \mathbf{I}$ 라고 가정한다. 이는 RGB 컬러 값은 서로 독립적이며 같은 분산 값을 가진다. 먼저, 각 분포의 계수를 초기화한 후 새로 들어온 프레임의 화소 X_j 에 대하여 각 분포의 표준 편차를 구하고, 표준 편차가 2.5이내인 매칭 분포를 찾은 후, 구한 분포를 w_k/σ_k 값이 큰 순서대로 정렬하여 이중 몇 개의 분포를 영상의 배경으로 간주하고 다음과 같이 계수를 갱신한다.

$$\begin{aligned} w_{k,i} &= (1-\alpha)w_{k,i-1} + \alpha M_{k,i} \\ \mu_{k,i} &= (1-\rho)\mu_{k,i-1} + \rho X_j \\ \sigma_{k,i}^2 &= (1-\rho)\sigma_{k,i-1}^2 + \rho(X_j - \mu_{k,i-1})^2 + (X_j - \mu_{k,i})^2 \\ \text{where} \\ M_{k,i} &= \begin{cases} 1, & \text{for the model which matched} \\ 0, & \text{for the remaining model} \end{cases} \\ \rho &= \alpha\eta(X_j | \mu_{k,i}, \sigma_{k,i}) \end{aligned} \quad (4)$$

여기서 망각 요인(forgetting factor) α 는 사용자가 정한 0과 1사이의 값을 가지는 실수로, 배경영상을 만들 때 이전 프레임의 영향을 조절하는 상수이다. 즉 이 값이 작으면 배경영상을 갱신할 과거 영상에 의한 관심이 커져서 과거 영상에 의한 배경 효과가 강하게 나타난다. 예를 들어, α 가 0.002인 경우 최근 500 프레임에 걸쳐 배경의 효과가 나타나므로 초기 배경을 생성하기 위해서는 500 프레임이 지나야 안정된 배경을 생성할 수 있다. 만약 매칭되는 분포가 없는 경우 w_k 값이 가장 작은 분포를 제거 한 후 새로 들어온 입력 값을 평균으로 가지며, 초기에 큰 분산과 아주 작은 w_0 를 갖는 새로운 분포를 삽입하여 계수를 갱신하여 배경을 추출하는 방법이다.

3. 벡터 순서 통계(Vector Order Statistics)를 이용한 배경 및 물체추출

본 논문에서 제안된 배경 및 움직이는 물체 추출은 그림 1과 같은 순서로 진행된다. 그림 1에서 프레임 샘플링이란 연속된 동영상 프레임에서 순차적으로 배경추출에 기여하는 프레임들을 추출하는 과정으로 계산량 절감과 안정적인 배경영상 취득에 도움이 된다. 또한 벡터정렬이란 샘플 프레임의 각 화소위치에서의 컬러 데이터를 정렬하는 과정이며, 각 화소위치에서 정렬이 끝나면 배경영상과 현재 프레임내의 물체영상을 추출하게 된다.



(그림 1) 제안된 배경영상 및 물체 추출과정

3.1 벡터 정렬

일반적으로 컬러영상의 한 화소는 3차원 벡터이다. 벡터의 순위결정을 위한 정렬 방법에는 여러 가지 방법이 제안되어 있으며[13-14], 제안된 알고리즘에서는 데이터 상호간의 거리 값 행렬에 의해 가장 작은 스칼라 값을 갖는 경우인 표본 데이터들의 가장 중심에 있는 데이터인 벡터 미디언으로 부터 외곽에 있는 데이터까지 효율적으로 정렬하는 방법을 이용하였다.

영상의 크기가 $M \times N$ 인 컬러영상의 각 화소에서 벡터 미디언은 벡터 공간에 나열된 데이터의 가운데 데이터로

$$\begin{aligned} \mathbf{x}_{jm} &= \{\mathbf{x}_j^i | i=1, \dots, N\} \\ \sum_{i=1}^N \|\mathbf{x}_{jm} - \mathbf{x}_j^i\|_2 &= \sum_{i=1}^N \|\mathbf{x}_j^k - \mathbf{x}_j^i\|_2, J=1, \dots, M \cdot N \\ & \quad \forall k=1, \dots, N \end{aligned} \quad (5)$$

과 같이 정의될 수 있다. 여기서 \mathbf{x}_{jm} 은 벡터 미디언을 의미하고, \mathbf{x}_j^i 를 i -번째 프레임의 j -번째 화소라 하자. $\mathbf{x}_j^i (i=1, \dots, N)$ 은 미디언 연산에 가담한 벡터들을 의미한다. 여기서 N 은 창의 크기로 프레임의 수를 나타낸다. \mathbf{x}_j^N 은 현재 프레임의 벡터이며, \mathbf{x}_j^1 은 현재 프레임을 기준으로 $N-1$ 번째 이전 프레임의 벡터이다. 각 벡터사이의 거리 행렬을 구하기 위하여 L_2 -norm을 사용하였으며, $\|\cdot\|_2$ 는 L_2 -norm을 나타낸다. 즉 벡터 미디언은 각 데이터 쌍의 거리들의 합을 최소로 하는 데이터이며, 이러한 벡터 미디언은 특이데이터에 강인한 대표치이며 컬러영상의 필터링, 또는 광 플로우의 평활화 등에 사용될 수 있다. 일반적으로 동영상에서 배경이 나타나는 빈도는 물체가 나타나는 빈도보다도 높기 때문에 해당 화소에 가장 빈번하게 나타났던 데이터 즉 벡터 미디언을 그 화소를 대표하는 배경값으로 가정할 수 있다.

또한 미디언 이외의 벡터는 식(5)에서 구한 벡터 미디언 \mathbf{x}_{jm} 을 중심으로 벡터사이의 스칼라 거리 값이 작은 순으로

$$\{\mathbf{x}_j^i | \mathbf{x}_j^1, \dots, \mathbf{x}_j^N\}, \mathbf{x}_j^1 = \mathbf{x}_{jm} \quad (6)$$

와 같이 정렬할 수 있으며, 여기서 $\mathbf{x}_j(k)$ 와 $\mathbf{x}_j(l)$ 사이의 관계는

$$\begin{aligned} \sum_{i=1}^N \|\mathbf{x}_j(k) - \mathbf{x}_j^i\|_2 &= \sum_{i=1}^N \|\mathbf{x}_j(l) - \mathbf{x}_j^i\|_2, \\ 1 \leq k \leq N-1, 2 \leq l \leq N, k < l \end{aligned} \quad (7)$$

과 같다.

3.2 배경 영상과 물체 영상의 추출

진술한 바와 같이 동영상에서 배경이 나타나는 빈도는 물체가 나타나는 빈도보다도 높기 때문에 해당 화소에 가장 빈번하게 나타났던 데이터 즉 벡터 미디언을 그 화소를 대표하는 배경값으로 가정할 수 있다. 또한 움직이는 물체는 빈번하게 발생하는 벡터 미디언으로부터 벗어난 위치에 있다. 즉 벡터 미디언으로부터 멀리 떨어져 있는 벡터들은 움직이는 물체로 간주 할 수 있다.

그림 1에서 벡터정렬은 현재 프레임에 포함된 샘플된 프레임들의 각 화소 위치에서 독립적으로 수행되며, 배경영상과 물체영상을 찾게 된다. 움직이는 물체를 추출하기 위해서는 전 질에서 언급된 정렬된 창내의 벡터들이 벡터 미디언으로부터 어느 정도의 거리 값으로 떨어져 있는지를 판별한다. 이 과정은 L_2 norm을 사용하여

$$d_j(t) = \| \mathbf{x}_j(1) - \mathbf{x}_j(t) \|_2, t = 1, \dots, N \quad (8)$$

와 같이 거리를 구하며, 각각의 벡터들이 얼마나 밀집 또는 분산되어 있는 지를 판단하기 위하여

$$T_j = \max_i \| d_j(i+1) - d_j(i) \|, i = 1, \dots, N \quad (9)$$

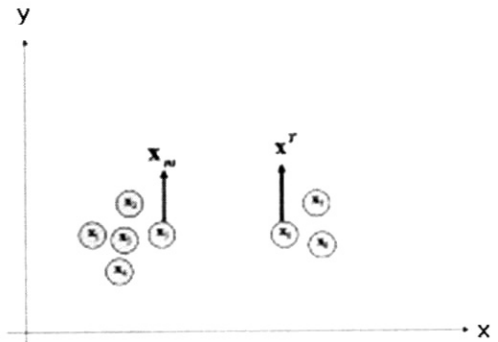
을 구한다. 여기서 T_j 는 벡터 미디언을 중심으로 순차적으로 거리가 증가하도록 벡터를 정렬하였을 때 거리값의 최대 증가분으로 벡터들의 분포를 알 수 있는 값이다. T_j 가 작은 값을 가지면 벡터분포는 벡터 미디언을 중심으로 밀집되어 있으므로 배경으로 간주 할 수 있으며, 큰 값의 T_j 는 벡터 미디언으로부터 멀리 떨어진 벡터가 하나 이상은 있다는 것을 나타낸다.

이러한 해석을 바탕으로 현재 프레임의 화소가 배경인지 움직이는 물체인지는

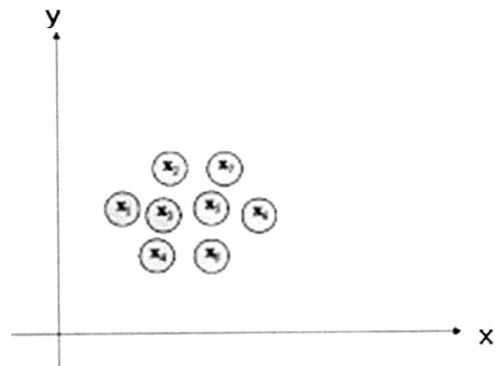
$$B(x) = \begin{cases} 0, & \text{if } \| \mathbf{x}_j(1) - \mathbf{x}_j^N \| \cdot \| \mathbf{x}_j(1) - \mathbf{x}_j^T \| \\ 1, & \text{otherwise} \end{cases} \quad (10)$$

로 판별한다. 여기서 \mathbf{x}_j^N 은 현재 프레임 입력이며, \mathbf{x}_j^T 은 식 (9)에서 T_j 값을 갖게 하는 벡터를 의미하며, $B(x)$ 는 x 위치에서의 물체여부를 가리키는 이진영상으로 $B(x)=1$ 인 경우가 물체 부분임을 의미한다.

그림 2는 2차원 벡터의 예를 보였다. 그림 2(a)는 2개의 군집으로 이루어진 8개의 벡터들을 표시하였다. 그림 2(a)에서 배경은 움직이는 물체보다 나타나는 빈도수가 높으므로 왼쪽 집단의 벡터들은 배경을 나타내고, 오른쪽은 물체를 나타낸다. 식 (5)에 의해서 벡터 \mathbf{x}_5 가 벡터 미디언이며, \mathbf{x}_6 은 식 (8)에 의해서 벡터 미디언과 분리된 벡터를 나타낸다. 만약 그림 2 a)에서 현재 프레임의 벡터가 $\mathbf{x}_6 \sim \mathbf{x}_8$ 의 그룹에 속하면 식 (9)에 의해서 움직이는 물체를 나타내며, $\mathbf{x}_1 \sim \mathbf{x}_5$



(a) T_j 값이 큰 경우



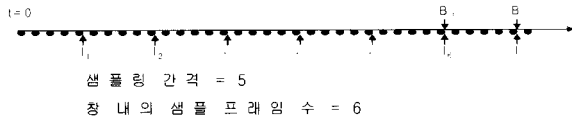
(b) T_j 값이 작은 경우

(그림 2) 창 내의 벡터들의 분포들

의 그룹에 속하면 그 화소는 배경으로 간주한다. 그림 2 (b)는 그림2 (a)와 달리 하나의 밀집된 벡터들을 나타내었고 배경으로만 구성되었다. 이럴 경우에 식 (9)의 T_j 값을 조절하여 하나의 분포로 간주하여 해당 화소를 배경으로 하였다.

3.3 프레임 샘플링 및 알고리즘 제어

진술한 바와 같이 모든 화소위치에서 벡터의 미디언을 찾는 데 필요한 계산량은 많은 계산량을 필요로 한다. 또한 매 프레임 새로운 배경을 갱신하는 것 역시 많은 계산량을 요하는 과정이다. 따라서, 본 논문에서는 배경영상의 갱신은 그림 1에서와 같이 샘플된 프레임에서만 일어나는 것으로 가정하였다. 그림 3은 시간축 상의 연속된 프레임에서 배경영상이 갱신되는 위치를 예로서 보여주고 있다. 그림 3에서는 샘플링 간격(W)은 5 프레임 중 한 장이 배경영상 갱신에 기여함을 의미하며, 최초 배경 B_1 을 만들기 위해서는 6장의 샘플된 프레임(S)이 필요함을 보여 주고 있다. 즉 최초 배경 B_1 은 $W \cdot S$ 프레임(그림3에서는 30 프레임) 후에 S장의 프레임의 매 화소 위치의 데이터들을 정렬하여 얻어지며, B_2 의 배경을 생성하는 시점에서는 I_1 프레임은 제거되며, I_2 부터 I_7 까지의 프레임을 이용한다. 따라서, 제안된 알고리즘에서 배경영상은 $W \cdot S$ 프레임 기간에 S 프레임을 이용하여 생성하며, 이러한 방법은 배경영상에 기여하는 프레임수를 증가시켜 물체가 천천히 움직이는 경우에도 물체의 흔적이



(그림 3) 프레임 샘플링과 배경경신

배경에 남지 않는 안정된 배경을 생성할 수 있다. 또한 벡터 정렬에 기여하는 프레임 수는 $W \cdot S$ 가 아니고 샘플된 S 개 이므로 계산량과 메모리 요구량을 감소할 수 있으며, 벡터의 거리 계산과 정렬에 있어서도 제거되는 프레임과 새로 추가되는 프레임만을 고려하여 계산량을 줄일 수 있다.

그러나, 위에서 언급한 방법을 적용하여 배경영상을 구하는 경우에도 물체영상은 매 프레임 계산된다. 즉 그림 3에서 생성된 B_1 배경 영상은 I_6 프레임의 물체를 검출하는데도 사용되며, B_2 이전의 프레임들의 물체를 검출하는데도 사용된다. 다시말하면 식 (10)에서 x_j^T 은 모든 현재 프레임의 의미하며, x_j^T 은 샘플된 S 개의 프레임에서 T_j 값을 갖게 하는 벡터를 의미하며, $x_j(1)$ 은 샘플된 S 개의 프레임의 미디언 영상 즉 배경영상을 의미한다. 물론 초기 $W \cdot S$ 프레임 동안은 배경영상은 생성되지 않으며, $W \cdot S$ 프레임이 진행된 후 최초의 배경영상과 물체검출이 시작된다.

따라서, 본 논문에서 제안한 알고리즘을 정리하면 다음과 같다.

- 1) Initialization : Set Step_Size(W), and Number_of_Samples(S)
- 2) Input new vector data of a frame
- 3) If it is a sample frame, then save it to Frame_Buffer Otherwise skip the frame
- 4) If Frame Buffer is Full, Find background image by taking vector median of equation (5), Arrange vectors according to distance from vector median to each vectors Calculate distances from vector median to each vector by equation (8) Find maximum distance(T) and the vector by equation (9) Continue from Step 5) Otherwise continue Step 2)
- 5) Classify input vector according to equation (10)
- 6) Input new vector data of a frame If it is a sample frame, Put it to Frame_Buffer and remove the last frame in the buffer Continue from Step 4) Otherwise continue from Step 5)

알고리즘에서 식 (9)에 의해서 구한 T_j 값은 벡터 분포의 밀집도를 나타내며 미리 설정한 문턱치보다도 작다는 것은 창 내의 벡터들이 모여 있으므로 발생 빈도수가 높아 해당 화소를 배경으로 간주 할 수 있다는 의미이다. 이 범위를 벗어나면 식 (10)에 의해서 입력 벡터가 어느 곳에 위치하는 지를 판별하여 분류 할 수 있다. 본 논문에서는 문턱치 값을 실험을 통하여 20으로 정하여 실험하였다.

4. 실험 및 고찰

본 논문에서 제안된 알고리즘의 성능평가를 위해 나뭇잎이 바람에 흔들리는 작은 도로를 책가방을 어깨 너머로 흔들며 지나가는 사람을 촬영한 영상과 서울 올림픽도로 차량 통행 장면을 촬영하여 실험하였다. 이 실험 영상들을 Lee[12]가 제안한 GMM을 알파만 변화시켜 가면서 실험하였고, 본 논문에서 제안한 벡터 정렬 방법은 베퍼 크기 및 샘플링 간격을 변화 시켜 가면서 실험하였다. 객관적인 평가 방법으로는 각각의 알고리즘으로 구한 배경 영상과 현재 프레임 영상의 차의 절대 값을 구한 후 같은 문턱치 값으로 이진 영상을 구하였다. 이렇게 얻어진 이진 영상을 레이블링하여 다음과 같은 계수를 구하였다.

$$M = (\text{검출 화소수}) / (\text{레이블 개수}) \tag{11}$$

레이블링은 구해진 이진 영상의 독립된 연결 객체에 대하여 레이블을 부여하는 과정이며, 사용된 레이블링 기법은 8-neighbor를 기반으로 하였다. 식 (11)의 M 값은 양호한 물체검출을 판단하는 척도로 만약 연결된 물체 성분이 많으면 M 값이 커지며, 나뭇잎이나 불안정한 물체 검출의 경우 연결 성분의 크기가 작아지거나 고립된 화소들이 물체로 검출되는 경우에는 M 값이 작아진다.

GMM 방법[9-11]은 현존 하는 방법 중에 반복적인 움직임 등에 비교적 잘 적용할 수 있는 모델로 많은 논문에서 실험되어 성능이 입증되었다. 본 논문에서는 Lee[12]가 제안한 GMM방법과 비교하였다. 이 방법은 최근에 발표되어 그간 문제되었던 GMM방법의 단점을 개선하여 좋은 결과를 얻었다. 그림 4 a)는 실험 동영상의 첫 번째 프레임을 보여주고 있으며 그림 4 b)는 155번째 프레임의 가방을 흔들며 지나가는 사람을 나타내고 있다.

그림 5와 6은 시간적 평균법과 시간적 중간치법을 이용하여 구한 배경 영상과 이진 영상을 나타내었다. 그림5와 6에서 구한 배경 영상은 창 크기를 20으로 하여 20개 프레임 화소의 평균값과 미디언 값을 구하여 나타내었다. 그림 5의 배경영상에 물체가 움직이고 난후 길다란 꼬리 모양과 같은



(a) 프레임 1



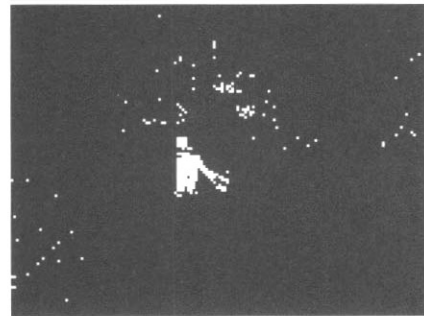
(b) 프레임 155
(그림 4) 실험 동영상



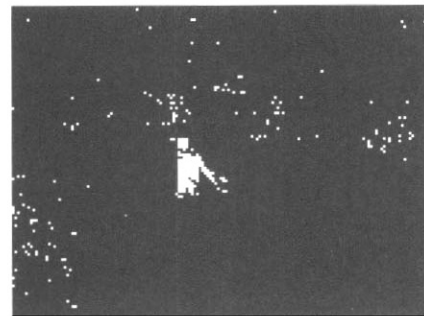
(그림 5) 시간적 평균법(창 크기 = 20)



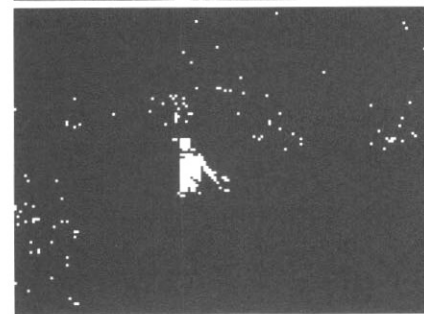
(그림 6) 시간적 중간치법(창 크기 = 20)



(그림 7) GMM을 이용한 배경영상과 이진영상($\alpha = 0.005$)



(a) 창 크기 = 20, 샘플링 간격 = 5



(b) 창 크기 = 30, 샘플링 간격 = 5

(그림 8) 제안한 알고리즘에 의한 배경영상과 이진영상

잔상이 발생함을 알 수 있다. 이는 배경영상 갱신에 최근 프레임의 물체 부분이 남아 평균치나 중간치에 영향을 주기 때문에 나타나는 현상으로 중간치 즉 미디언을 이용하여 배경영상을 취득하기 위해서는 물체가 점유하는 기간보다 배경이 점유하는 기간이 길 정도로 창 크기를 크게 설정하여야 안정된 배경영상을 취득할 수 있음을 의미한다.

그림 7은 GMM 방법을 통하여 얻어진 계수를 이용하여 구한 배경영상과 이진영상을 나타내었다. 그림 7의 배경영상은 155번째 프레임까지 처리된 계수를 이용하여 구한 것으로 망각 요인(α)를 0.005로 하여 구하였다. 그림 7에서 알 수 있듯이 GMM을 이용한 배경영상은 반복적으로 움직이는 나뭇잎이나 이동하는 물체의 영향을 받지 않고 깨끗한 배경을 추출하였다.

그림 8은 본 논문에서 제안한 벡터 미디언을 이용하여 구한 배경영상 및 이진영상을 나타하였다. 그림 8에서 알 수 있듯이 창 크기가 커짐에 따라 이진 영상의 물체의 표현이 점점 정확해 지며, 움직이는 물체에 대한 배경영상도 양호해졌다.



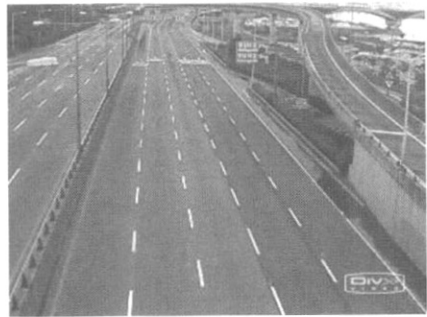
(a) 200번째 프레임



(b) GMM-200($\alpha=0.005$)



(c) GMM-200($\alpha=0.01$)



(d) 벡터 정렬 방법-200
(창크기=20, 샘플링간격=5)



(e) 벡터 정렬 방법-200
(창크기=10, 샘플링간격=10)



(f) 벡터 정렬 방법-200
(창크기=20, 샘플링간격=10)

(그림 9) 올림픽 도로(200 프레임) - 배경영상들

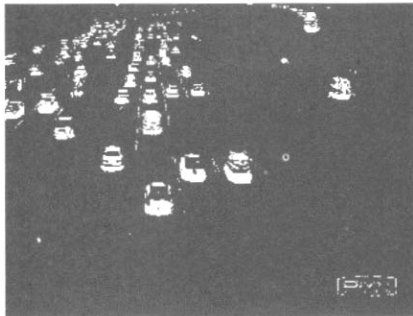
그림 9는 올림픽 도로 상황을 촬영한 실험 영상으로 (a)는 200번째 프레임을 나타내었으며 나머지는 GMM과 본 논문에서 제안한 방법을 창 크기와 샘플링 간격을 변화시켜 구한 배경영상이다.

그림 9의 b)는 알파가 0.005이므로 적어도 계수가 안정되기 위해서는 200프레임 이상 갱신 후 안정된 배경영상을 얻을 수 있다. 이런 이유로 중앙 분리대를 중심으로 1차선에 있는 차량의 잔상이 아직 남아있다. 본 논문에서 제안한 벡터 정렬 방법은 창 크기 및 샘플링 간격에 따라 구한 배경영상이 상이함을 알 수 있다. 창 크기 = 20, 샘플링 간격 = 10 일때 가장 좋은 배경영상을 추출할 수 있다.

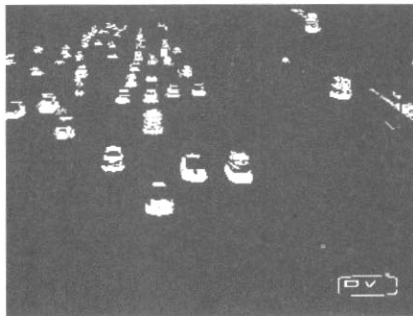
그림 10은 그림 9에서 각각의 알고리즘으로 구한 배경영상과 현재 영상의 차를 구한후 절대값을 취하여 문턱치를 20으로 하여 이진 영상을 구하였다. 표 1은 얻어진 이진 영상을 식 (9)를 이용하여 평가척도를 구하였다. 본 논문에서 제안한 방법이 GMM 방법보다 양호함을 알 수 있다.

<표 1> 알고리즘 성능 평가 - 200 번째 프레임

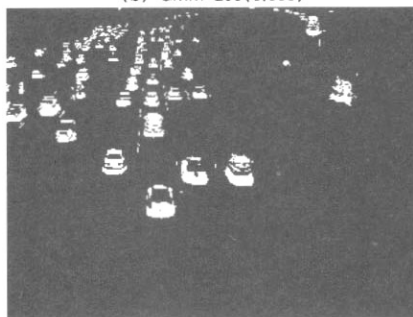
Frame Number	Method					비고
		pixels	label	pixels/label	평가 순위	
200	GMM	3343	147	22.7	5	0.01 (Alpha)
	GMM	3741	159	23.5	4	0.005 (Alpha)
	벡터 정렬 방법	3445	131	26.3	2	20-5 (Buffer size-Sampling)
	벡터 정렬 방법	3411	133	25.6	3	10-10 (Buffer size-Sampling)
	벡터 정렬 방법	3628	135	26.9	1	20-10 (Buffer size-Sampling)



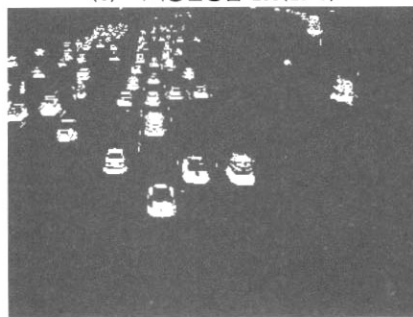
(a) GMM-200(0.01)



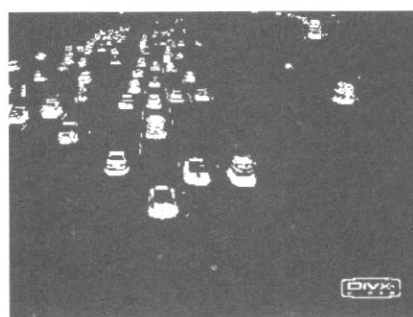
(b) GMM-200(0.005)



(c) 벡터정렬방법-200(20-5)



(d) 벡터정렬방법-200(10-10)



(e) 벡터정렬방법-200(20-10)

(그림 10) 올림픽 도로(200 프레임) - 이진 영상들

<표 2>는 500 프레임 평균을 구하여 나타내었다. <표 2>에서 알 수 있듯이 창 크기 = 20 샘플링 간격 = 10일때 가장 좋은 결과를 얻었다.

<표 2> 알고리즘 성능 비교 - 500 프레임 평균

Method	Parameter	pixels/label	평가순위
GMM	0.01(alpha)	35.08	3
GMM	0.005(alpha)	34.29	5
벡터 정렬 방법	20-5	36.73	2
벡터 정렬 방법	10-10	34.78	4
벡터 정렬 방법	20-10	37.95	1

이상의 실험을 통해 시간적 중간치 방법을 개선한 본 논문에서 제안된 알고리즘은 다음과 같은 성질을 가짐을 알 수 있다. 먼저 샘플링 간격과 창의 크기는 동영상 프레임 내에서 이동 물체가 연속되는 프레임에 나타나는 기간보다 배경이 점유하는 기간이 큰 범위에서 결정되어야 한다. 즉 미디언을 배경으로 가정하기 때문에 미디언을 결정하기 위한 다수의 화소가 배경을 점유하는 화소 이어야 안정된 배경을 생성할 수 있다는 의미이다. 또한 샘플링 간격은 프레임 버퍼 내의 배경이 충분히 적응적일 수 있을 만큼 짧아야 한다. 즉 샘플링 간격 내에서의 배경변화가 없어야 한다는 의미이다.

GMM의 경우와 제안된 방법의 차이점은 전자의 방법에서는 충분한 프레임 수에 의해 모델의 안정성이 보장되어야 안정된 배경을 추출할 수 있는데 반하여 제안된 방법에서는 초기에 프레임 버퍼 즉 창 크기의 프레임이 지난 후면 안정된 배경과 물체 구분이 가능해 진다는 것이다. 즉 모델의 안정성이 보장하기 위한 초기의 충분한 샘플을 가정하지 않아도 미디언이 배경으로 보장되는 만큼의 W*S 프레임이 확보된 후에는 안정된 배경과 물체 검출이 가능하다는 것이다.

제안된 알고리즘의 계산량은 만약 샘플링을 전제로 하지 않을 경우에는 GMM의 경우보다 많으나 샘플링을 전제로 하는 경우에는 소수의 샘플 (약 20 프레임)에서 채귀적으로

벡터 미디언 계산이 이루어지기 때문에 GMM 방법에 비해 크게 증가하지는 않는다. 또한 메모리 요구량은 GMM 방법의 경우 매 화소당 평균과 공분산 행렬을 저장하는 것에 비해 화소값과 화소간의 거리 값들을 저장하여야 하기 때문에 증가할 수밖에 없다.

5. 결 론

배경 추출은 비디오 감시 시스템에서 움직이는 물체를 찾는 데 중요한 부분이다. 본 논문에서는 벡터 정렬법을 이용한 새로운 배경영상 추출 방법을 제안하였다. 제안된 방법에서 배경은 물체보다 발생 빈도가 높다는 사실에 의하여 창 내의 벡터를 거리에 의한 벡터 미디언을 찾고, 이 벡터 미디언을 중심으로 창 내의 나머지 벡터를 정렬하여 연속되는 프레임에서 배경과 물체를 구별하였다. 제안된 알고리즘은 기존의 시간적 중간치법을 개선한 것으로 시간적 중간치 방법보다 샘플링에 의해 계산량이 줄어들며, 안정된 배경과 물체 검출이 가능하다.

제안된 알고리즘의 성능을 평가하기 위하여 가우시안 혼합 모델을 이용한 다중 배경 방법과 연결된 물체성분 당 화소의 개수를 비교한 결과 제안된 벡터 정렬 방법이 유사 또는 양호함을 알 수 있었다. 향후 배경의 변화 속도 및 물체의 이동 속도 등 비디오 환경에 따른 버퍼크기 및 샘플링 간격을 정량적으로 연구하여 제시할 것이다.

참 고 문 헌

[1] I. Haritaoglu, D. Harwood and L. S. Davis, "W4:Real Time Surveillance of people and Their Activities," IEEE Trans. PAMI, Vol.22, No.8, pp.809-830, 2000.

[2] 조태훈, 최영규 "다중 배경 분포를 이용한 움직임 검출", 정보처리학회논문지 제8-B권 제4호, pp.381-388, 2001.

[3] N. Frieman and S. Russell, "Image segmentation in video sequences: A probabilistic approach," Proc. 13th Conf. Uncertainty in Artificial Intelligence, Aug., 1997.

[4] D. Wang, "A Novel Probability Model for Background Maintenance and Subtraction," 15th Conf. Vision Interface May, 2002.

[5] S. S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," Proc. of SPIE : Visuual Communications and Image Processing, 2004.

[6] Z. Hou and C. Han, "A Background Reconstruction Algorithm based on Pixel Intensity Classification in Remote Video Surveillance," Proc. of 7th Conf. Infor-

mation Fusion, June, 2004.

[7] B.P.L. LO and S.A. Velastin, "Automatic Congestion Detection System for Underground Platforms," Proc. Symp. Intelligence Multimedia, Video, and Speech Processing, pp.158-161, 2000.

[8] B. Gloyer, H.K. Aghajan, K. Y. Siu and T. Kailath, "Video-Based Freeway Monitoring System Using Recursive Vehicle Tracking," Proc. SPIE Symp. Electronic Imaging: Imag and Video Processing, 1995.

[9] R. Cucchiara, C. Grana, M. Piccardi and A. Prati, "Detecting Moving Objects, Ghosts, and Shadows in Video Streams," IEEE Trans. PAMI, Vol.25, No.10, October, 2003.

[10] C. Stauffer and W. E. L. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," Proc. Computer Vision and Pattern Recognition 99, Colorado.

[11] P.KaewTraKulPong and R. Bowden. "An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection," 12nd. European Workshop on Advanced Video Based. Surveillance Systems, AVBS01, September, 2001.

[12] D. S. Lee. "Effective Gaussian Mixture Learning for Video background Subtraction," IEEE Tran. on PAMI, Vol.27, pp.827-832, 2005.

[13] J. Astola, P. Haaristo and Y. Neuvo, "Vector Median Filters," Proceeding of IEEE, Vol.78, No.4, pp.678-689, 1990.

[14] 엄경배, 한서원, 이준환, "혼합된 컬러 잡음하에서 컬러 영상 향상을 위한 조건적인 퍼지 클러스터 필터" 정보처리학회논문지 제6권 제12호, pp.3718-3726, 1999.

김 준 철



e-mail : jckim@seonam.ac.kr

1986년 전북대학교 전자공학과(공학사)
 1988년 전북대학교 전자공학과(공학석사)
 1988년~1991년 LG산전연구소 연구원
 1995년 전북대학교 전자공학과(공학박사)
 1993년~현재 서남대학교 전지전자공학과
 조교수

관심분야: 영상처리 및 컴퓨터비전

박 은 종



e-mail : for0511@paran.com
2003년 군산대학교 컴퓨터정보학과
(학사)
2005년 전북대학교 전자공학과 업(석사)
2005년~현재 전북대학교 영상공학과
박사과정

관심분야: 영상처리, 컴퓨터비전

이 준 환



e-mail : chlee@chonbuk.ac.kr
1980년 한양대학교 전자공학과(공학사)
1982년 한국과학기술원 전자공학과(공학석사)
1982년~1985년 전북대학교 전자공학과
조교
1985년~1987년 전북대학교 전자공학과
전임강사

1990년 미주리대학 전산학과(공학박사)
1990년~현재 전북대학교 전자정보공학부 교수
관심분야: 영상처리 및 분석, 인공지능