논문 04-29-8C-17 한국통신학회논문지 '04-8 Vol 29 No 8C

Mean-Shift 알고리즘을 이용한 MPEG2 압축 영역에서의 움직이는 객체 추적

박 성 모*, 정회원 이 준 환**

Tracking of Moving Object in MPEG Compressed Domain Using Mean-Shift Algorithm

Seong-Mo Park*, Joonwhoan Lee** *Regular Member*

요 약

본 논문에서는 MPEG2비디오 스트림에서 복호화 과정없이 압축비디오에서 직접 얻은 수 있는 정보들을 활용하여 움직이는 객체를 추적하는 방법을 제안한다. 제안된 방법에서는 먼저 MPEG2의 움직임 벡터로부터 근사적으로 움직임 플로우(motion flow)를 구성하고, 전역적인 움직임 플로우로 부터 일반화된 Hough 변환을 이용 카메라의 기본적인 움직임인 팬(pan), 틸트(tilt), 줌(zoom)량 등을 계산하였다. 계산된 카메라 움직임은 국부적으로 일어나는 객체의 움직임을 보정하는데 사용하였다.

움직이는 객체의 추적은 사용자가 원하는 객체를 바운딩 박스 형태로 정의함으로 시동된다. 이후의 객체의 추적은 카메라 움직임이 보정된 객체의 움직임 플로우를 이용하여 Mean-Shift 알고리즘을 이용하여 추적하였다.

제안된 방법은 압축된 비디오 스트림에서 직접 정보를 얻음으로써 계산속도의 향상을 기할 수 있으나, 압축된 MPEG2 비디오에서 얻을 수 있는 정보들이 최대 블록 단위이므로 객체의 정의도 블록단위 이상의 객체로 제한된다.

주요어 : MPEG-2, 비디오 스트림, 압축 도메인, Mean-Shift 알고리즘, 객체추적

ABSTRACT

This paper propose a method to trace a moving object based on the information directly obtained from MPEG-2 compressed video stream without decoding process. In the proposed method, the motion flow is constructed from the motion vectors involved in compressed video and then we calculate the amount of pan, tilt, zoom associated with camera operations using generalized Hough transform. The local object motion can be extracted from the motion flow after the compensation with the parameters related to the global camera motion.

The moving object is designated initially by a user via bounding box. After then automatic tracking is performed based on the mean-shift algorithm of the motion flows of the object.

The proposed method can improve the computation speed because the information is directly obtained from the MPEG-2 compressed video, but the object boundary is limited by blocks rather than pixels.

Key Words MPEG-2, Video stream, Compressed Domain, Mean-Shift Algorithm, Object Tracking

* 전북대학교 전자정보공학부 인공지능 연구실(digboy@hanmail.net), ** 전북대학교 전자정보공학부(chlee@moak.chonbuk.ac.kr)
논문번호 030293-0715, 접수일자 2003년 7월 15일
※본 연구는 대학 IT 연구센터 육성지원 사업의 연구결과로 수행되었습니다.

Ⅰ. 서 론

컴퓨터의 발달과 인터넷의 확산으로 멀티미디어 콘텐츠의 보급이 급속히 확대되고 있다. 이들 콘텐츠에는 원거리 화상회의, 감시시스템, 주문형 비디오(VOD), 주문형 뉴스(NOD), 디지털 편집 시스템 등 동영상이 포함되어 있다. 그러나 이와 같은 동영상 서비스를 한정된 대역폭을 통해 저장하거나 전송하기 위해서는 동영상이 가지는 방대한 양의 데이터를 MPEG과 같은 방식으로 압축하는 것이 필수적이다.

본 논문에서는 이와 같이 MPEG으로 압축된 데이터에 포함된 정보를 효율적으로 이용하여, 검색 및 그와 관련된 데이터 베이스의 구축에 활용되는 움직이는 객체의 추적 및 해석을 다루고자 한다. 이러한 목적을 위해서 기존의 방법들에서는 압축된 동영상 데이터를 복호화 한 후 시공간 영역에서 처리[1]하였는데 다양한 정보를 비디오로부터 얻을 수 있다는 융통성이 있는 반면에 속도저하는 물론, 부호화 과정에서 발생되는 유용한 정보 즉 움직임 정보나 블록의 질감 정보 등의 간과하고 다시 계산될 수 있다는 단점을 내포하고 있다.

이러한 단점을 극복하고자 압축영역에서의 압축에 수반된 다양한 정보를 영상분석에 이용하고자 시도하였는데, S-F Chang 등은 DCT (Discrete Cosine Transform) 영역에서 질감을 분석하는 여러 방법을 제안하였다[2]. 또한 V. Kobla는 MPEG의 움직임 벡터를 이용하여 압축영역에서 움직임 플로우 개념을 확립하였으며[3], A. Yoneyama와 Y. Nakajima는 압축영역에서 움직이는 객체를 정의하고 해석하였으며[5], Favalli 등은 객체 추적에 움직임 벡터를 이용할 수 있음을 보였다[8].

그러나 객체의 움직임 벡터와 객체의 질감만으로 움직이는 객체를 추적하는 것은 카메라의 움직임이 있는 경우 카메라의 움직임과 객체의 움직임 벡터가 혼재하고, 원래 MPEG의 움직임 벡터가 부호화 효율을 증가시키기 위해 계산되었기 때문에[4] 그대로 객체의 움직임 플로우로 간주하는 데는 여러 가지 문제점을 가지고 있다. 후자의 문제는 매크로 블록 단위의 움직임 벡터로부터 움직임 플로우를 계산하고 부호화 효율 때문에 잡음형태로 발생하는 플로우를 시공간 미디언 필터를 이용하여 제거함으로 해결할 수 있다. 또한 카메라의 움직임이 전역적인데 반하여 객체의 움직임은 국부적인 사실을 이용하여 단순한 카메라 모델을 정의하고 부수되는 파라메터를 일반화된 Hough 변환을 이용하여 카메라 움직임을 추정하고, 이를 이용하여 객체만의 움직임을 추정할 수 있다.

본 논문에서는 이러한 점을 고려하여 움직임 벡터로부터 계산된 움직임 플로우에서 카메라의 움직임을 보상한 후 매크로 블록 당 국부적인 객체의 움직임을 계산하고, 이를 이용하여 사용자에 의해 미리 정의된 바운딩 박스내의 객체를 추적하는 방법을 제안하였다. 제안된 방법에서는 움직임 플로우의 분포와 mean-shift 알고리즘을 이용하여[11], 연속되는 프레임에서 추적객체를 재 설정하였다. 본 논문의 제안된 방법은 Favalli 등의 방법에서 움직임 벡터를 사용한 추적과 유사하나, 카메라 움직임 보상을 수행한 후에 객체의 추적이 진행되기 때문에 카메라의 움직임과는 무관하게 객체의 추적이 가능하며, mean-shift 알고리즘에 의해 불필요한 탐색이 줄어들기 때문에 빠른 추적이 가능하다.

본 논문의 2절에서는 MPEG 비디오 스트림으로부터 객체의 움직임을 추출하는 방법을 기술하였으며, 3절에서는 움직이는 물체추적을 위한 mean-shift 알고리즘을 설명하였고, 4절에서는 구현된 시스템 및 실험결과에 대해서 논하였다. 마지막으로 5절에서는 본 논문의 결론 및 확장방법에 대해서 논하였다.

Ⅱ. 객체의 움직임 추출

본 논문에서 제안된 객체 추적알고리즘은 압축된 MPEG 비디오에 담겨있는 정보를 복호화 과정없이 직접 매크로 블록당 움직임을 추출한다. 그림 1은 MPEG 비디오 스트림에서 객체의 움직임 플로우를 추출하는 과정을 보여주고 있다. 그림에서 알 수 있듯이 객체의 움직임을 추출하는 과정은 MPEG-2 비디오 스트림에서 움직임 플로우를 추정하고, 추정

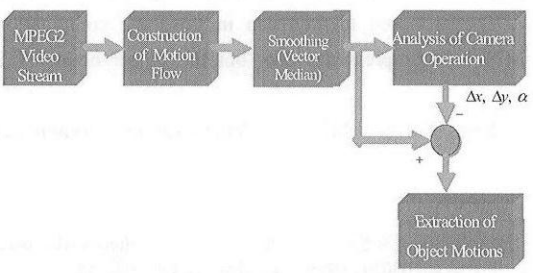


그림 1. 객체 움직임의 추출

된 움직임 플로우를 시공간 벡터 미디안 필터를 이용하여 평활화하며, 카메라 동작을 보상하여 얻어진다.

2.1 근사적인 움직임 플로우의 구성

MPEG 비디오의 움직임 벡터는 압축효율을 높이기 위해 사용되기 때문에 이들 성분들이 순서적이지 못하고, 공간적으로 균질(homogenous)한 영역에서는 움직임 벡터들이 불규칙하여 카메라 또는 객체의 움직임을 잘 반영하지 못한다.[4] 또한 모든 매크로 블록이 움직임 벡터를 가지는 것은 아니다. 때문에 움직임 벡터들로부터 근사적인 움직임 플로우를 계산하기 위해서는 먼저 각 프레임의 압축 방법에 따라 또는 움직임 벡터의 성질에 따라 순서적으로 재배열하고, 배열된 벡터들을 필터링하여 난잡한 움직임을 평활화하고 움직임이 정의되지 않은 블록이 가능한 적어지도록 하는 보간 과정이 필요하다.

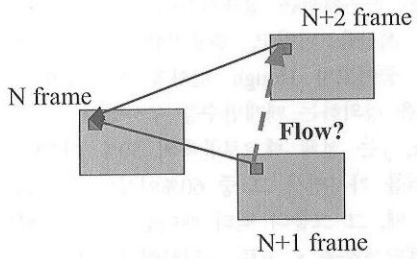


그림 2. 매크로 블록단위의 움직임 플로우 예측

그림 2는 매크로 블록단위의 움직임 벡터로부터 움직임 플로우 예측의 필요성을 보여주고 있다. 그림에서 N 프레임은 참조 프레임(reference frame)을 표현하며 N+1, N+2 프레임이 참조 프레임을 참조하여 실선과 같은 움직임 벡터를 가진다고 가정할 경우 참조 프레임에서 N+1 프레임으로의 움직임 플로우는 움직임 벡터(실선)의 반대방향으로 움직임 벡터와 같은 크기로 가정할 수 있다. 그러나, N+1 프레임에서 N+2 프레임으로의 움직임 플로우는 움직임 벡터로부터 직접 얻어낼 수 없고 두 움직임 벡터의 차를 이용할 수밖에 없다.

움직임 플로우의 예측을 위해 프레임의 연결에 따라 고려할 형태는 다음의 4 종류가 있다[5].

(a) I(t) + P(t+n) 또는 P(t) + P(t+n)

(b) B(t) + B(t+1)

(c) B(t) + I(t+1) 또는 B(t) + P(t+1)

(1)

(d) I(t) + B(t+1) 또는 P(t) + B(t+1)

식 (1-a)에서의 경우는 그림 3에 표현된 바와 같다. 그림 3의 첫 번째 경우는 순방향 예측된 움직임 벡터의 반대방향을 움직임 플로우로 가정할 수 있는 경우를 의미하고, 두 번째 경우는 두 참조 프레임 사이에 B 프래임이 삽입된 경우로 움직임 벡터의 크기를 삽입된 프레임수+1로 나누고 방향을 반대로 표현하는 경우이다.

그림 4는 (1-b)의 경우로 움직임 벡터의 종류에 따라 (순방향, 순방향), (역방향, 순방향), (역방향, 역방향)인 경우에 각 움직임 벡터의 차이에 의해 움직임 플로우를 구성할 수 있다. 그러나, 그림 4의 마지막에 표현된 (순방향, 역방향)인 경우에는 움직임 플로우를 구성할 수 없다.

식 (1-c)의 경우에는 B 프래임이 참조한 프래임의 움직임 벡터는 방향 전환 없이 그대로 움직임 플로우로 가정하였으며, 식 (1-d)의 경우에는 B 프레임의 순방향 움직임 벡터의 반대방향을 움직임 벡터로 가정하였다.

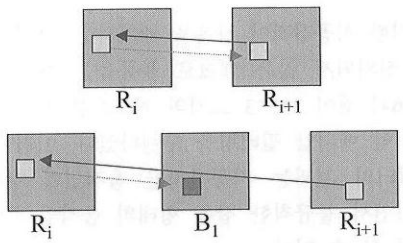


그림 3. 참조 프래임간의 움직임 플로우 추정

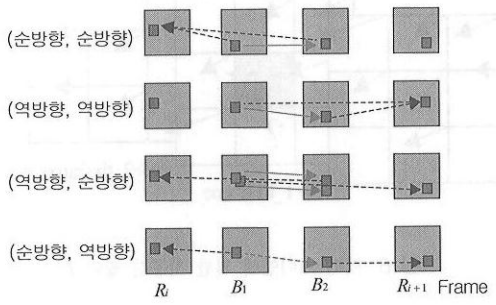


그림 4. B 프래임간의 움직임 플로우 추정

또한 식 (1)에서 예시한 움직임 플로우를 생성하는 과정에서 움직임 벡터의 역방향 성분에 따라 해당 매크로 블록을 이동하면 참조 프레임의 매크로 블록과 일반적으로 일치하지 않는다. 이러한 경우에는 참조 프레임에 가장 많이 겹치는 그림 5의 매크로 블록이 해당 움직임 벡터의 역방향의 움직임 플로우를 갖는다고 가정하였다.

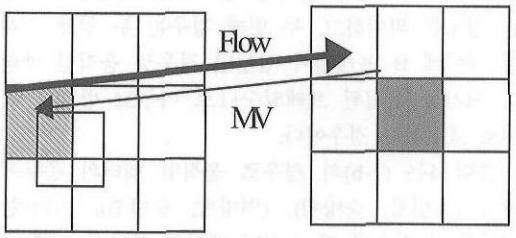


그림 5. 움직임 벡터와 역 방향인 움직임 플로우의 구성

2.2 움직임 플로우의 보간 및 평활화

전술한 방법으로 구성된 매크로 블록단위의 움직임 플로우는 MPEG 비디오가 압축향상을 목적으로 움직임 벡터를 예측하고 부호화 하기 때문에 시공간적으로 많은 잡음을 포함하고 있으며, 움직임 플로우가 정의되지 않는 매크로 블록을 포함할 수 있다.

이러한 시공간적인 잡음을 제거하고, 움직임 플로우가 정의되지 않은 매크로 블록을 보간하기 위해 그림 6과 같이 3*3*3 크기의 윈도우를 이용하여 시공간적인 메디안 필터링을 수행하였다. 이러한 메디안 필터의 결과는 시공간적인 움직임의 불연속을 보존하면서 불규칙한 잡음 형태의 움직임 플로우를 평활화 할 수 있다.

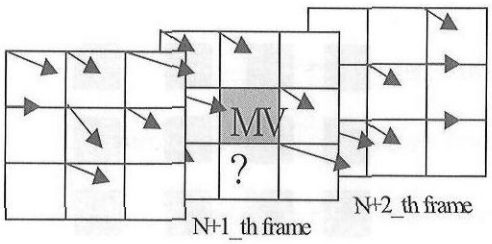


그림 6. 벡터 미디안에 의한 시공간 필터링

그림 6의 필터링 대상은 움직임 플로우에 해당하는 벡터정보로서 메디안 벡터는 해당 윈도우 내의 광 플로우 벡터간의 유클리디안 거리의 합을 최소로 하는 벡터에 해당한다. 이러한 총체화 정렬(aggregate ordering)방식을 이용한 벡터정보의 순서통계(order statistics) 필터링은 벡터의 불연속을 보존하며 가우시안 형태의 잡음을 효과적으로 제거할 수 있다.

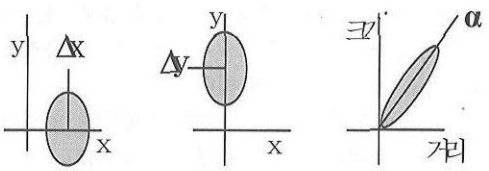
2.3 카메라 동작의 추정

카메라의 움직임은 어파인(affine) 모델과 원근투영(perspective-projection) 모델 등을 이용하여 근사화 할 수 있다. 그러나 이러한 접근은 압축정보에서 구할 수 없는 카메라의 초점거리를 사용해야 하므로 본 논문에서는 가장 단순한 팬-틸트-줌(pan-tilt-zoom) 모델을 사용했다[6][7].

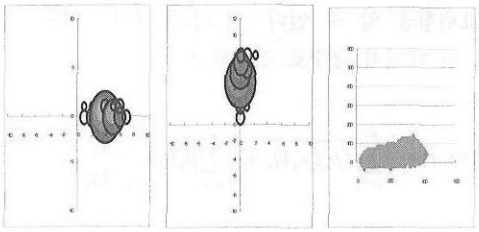
이러한 모델에서 R.Milanese등은 프레임의 대표값인 x, y, 을 구하기 위해서 사용하였는데[6], 이들은 최소자승(least-mean square) 알고리즘을 반복적으로 사용하여 프레임의 대표 값들을 구하였다. 그러나 본 논문에서는 실행시간이 많이 요구되는 반복적인 계산을 피하고, 불규칙한 움직임 플로우의 영향을 줄이고자 Hough 변환을 이용하여 카메라 움직임을 정의하는 매개변수를 구하였다.

즉 x, y는 전체 매크로블록의 50% 이상이 움직임 벡터를 가지면서 그 중 60%이상이 한 그룹으로 포함될 때, 그 그룹의 벡터 메디안으로 대표되는 움직임 벡터성분을 x, y로 정의하였다. 그리고 는 줌의 중심좌표를 중점으로 가진 원을 기준으로 밖으로 향하는 성분크기를 양의 값, 안쪽으로 향하는 성분의 크기를 음의 값을 가진 크기와 중심부터의 거리로써 Hough 공간을 정의하고 가장 많은 분포를 대표하는 기울기를 로 정의하였다.

그림 7의 (a)는 카메라가 팬, 틸트, 줌 동작이 발생시 이상적인 매개변수의 분포도이다. 또한 (b) 는 실제영상에서 구한 매개변수의 분포도이다.



(a) 이상적인 매개변수 분포도



(b) 실제 영상에서의 매개변수 분포도

그림 7. 팬 틸트 줌의 매개변수 분포도

카메라 동작을 정의하기위한 x, y, 의 매개변수는 매 프레임마다 계산 되어지며, 연속적인 카메라 움직임을 추정할 수 있다. 또한 값을 계산하기위한 중심좌표는 매개변수 값을 결정하는데 중요한 요인이 되는데 측정에러를 줄이고자 각 프레임마다 가로와 세로방향으로 움직임 플로우의 y와 x성분이 없는 매크로 블록들에게 가중치를 주어, 누적하여 중심좌표를 구하는 방법을 취하였다. 그림 8은 확대(zoom-in)가 발생할 때 각 매크로 블록들의 움직임 벡터의 상태와 가중치를 표현하고, 가장 많이 누적된 중심의 x 좌표를 찾는 그림의 예이다. 그림에서 움직임이 정의되지 않은 매크로 블록의 x 좌표들을 그림 우측의 가중치와 곱하여 행(column) 방향의 합(sum)하고 이들 합 중 최대의 합을 갖는 행(column)을 찾아 해당 매크로 블록 구간의 중앙을 x 방향의 중심좌표로 설정하였다. 줌 중심의 y좌표도 이와 동일하게 계산된다.

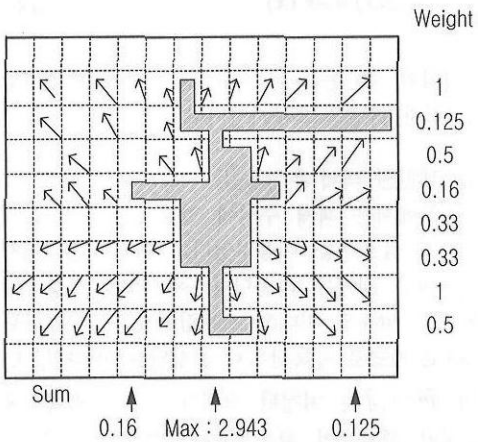


그림 8. 줌의 중심좌표 중 x성분 구하는 예

2.4 카메라 동작을 배제한 움직임 추출

MPEG 비디오의 움직임 벡터는 카메라 동작과 객체의 움직임이 혼재되어 나타난다. 즉 카메라의 동작에 의한 움직임이 프래임 전 영역에 대해 나타난다면 객체의 움직임은 국부적인 영역에 나타난다. 따라서 움직임 벡터로부터 구성한 움직임 플로우의 경우도 같은 성질을 가지고 있으며, 객체의 움직임을 추적하기 위해서는 전절에서 구한 카메라의 움직임을 이용하여 보상되어야 한다. 즉 팬과 틸트인 경우 객체를 제외한 배경영역의 블록이 팬과 틸트의 카메라 동작에 해당되는 x, y 값의 움직임 벡터를 가지게 된다. 그러므로 움직임 플로우에서 팬과 틸트에 해당되는 카메라 동작에 의한 움직임을 제거할 수 있다. 또한 줌의 경우 매크로 블록당 움직임 플로우에서 확장 또는 축소정도를 표현하는 값(기울기)에 따라 움직임 플로우의 값을 빼면 카메라 동작을 제거한 순수한 움직이는 객체의 움직임을 추정할 수 있다. 그림 9는 팬, 틸트와 줌이 발생하는 부분에서 카메라 동작에 의한 움직임 플로우를 보상하고, 움직임 플로우가 있는 부분, 즉 추적대상이 될 수 있는 객체일 가능성이 있는 블록을 보여주고 있다.

그림 9. 팬과 줌의 경우 카메라 동작을 보상한 후의 추적대상 객체

Ⅲ. Mean-Shift 알고리즘을 이용한 객체 추적

본 논문에서 제안된 방식에서 객체의 추적은 Mean-Shift 알고리즘을 이용한다.[9][10]

Fukunaga에 의해 제안된 Mean-Shift 알고리즘은 일반화된 클러스터링 알고리즘으로 객체의 세그멘테이션, 필터링, 최적화 문제의 해결 등에 유용하게 사용되었다[9]. 본 논문에서는 Meer등에 의해 제안된 mean-shift를 이용한 객체추적 알고리즘에서 칼라분포 대신에 객체영역의 움직임 플로우의 분포를 적용하였다.

3.1 Mean-Shift를 이용한 최적화 및 객체추적 방법[11]

Mean-shift 알고리즘은 일련의 확률분포의 최고치를 최대경사법(gradient ascent method)을 기반으로 한 반복적인 계산에 의해 추정할 수 있다. Mean-Shift 알고리즘을 이용한 객체추적은 두 히스토그램사이의 Bhattacharyya 측도

$$\hat{\rho}(y) \equiv \rho[\hat{p}(y),\hat{q}] = \sum_{u=1}^{n}\sqrt{p_u(y)q_u} \tag{2}$$

를 최대화하는 과정으로부터 얻어질 수 있다. 식 (2)에서 m은 히스토그램의 빈의 개수를, $\hat{p}(y)$는 y를 중심으로 한 속성의 정규화된 히스토그램으로

$$\hat{p}_u(y) = C_h\sum_{i=1}^{n_h} k\left(\left\|\frac{y-x_i}{h}\right\|^2\right)\delta[b(x_i)-u] \tag{3}$$

와 같이 표현되며, $\hat{q}$는 기준이 되는 히스토그램으로 [12]

$$\hat{q}_u = C'\sum_{i=1}^{n} k\left(\left\|x_i^*\right\|^2\right)\delta[b(x_i^*)-u] \tag{4}$$

와 같다. 식 (3)과 (4)에서 $k()$는 공간영역에서의 커널함수를 표현하며, y, x_i들은 각각 공간영역의 좌표들로, 파라메터 h에 의해 스케일링되며, $\delta[]$는 dirac 함수로 x_i, x_i^*등에서의 인덱스 추출함수 $b(x_i), b(x_i^*)$들의 값이 해당 인덱스 변수 u와 같을 때만 1의 값을 갖는다. 즉 식 (3)과 (4)에서 C와 C_h가 정규화를 위한 상수일 경우 덧셈감소하는 공간영역의 커널함수로 가중화된 정규화된 히스토그램을 표현함을 알 수 있다. 식 (2)를 $\hat{p}_u(y_0)$를 중심으로 한 테일러 급수로 전개하면

$$\rho[\hat{p}(y),\hat{q}] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u(y_0)\hat{q}_u} + \frac{1}{2}\sum_{u=1}^{m}\hat{p}_u(y)\sqrt{\frac{\hat{q}_u}{\hat{p}_u(y_0)}} \tag{5}$$

와 같으며, 식 (3)을 대입하여 정리하면

$$\rho[\hat{p}(y),\hat{q}] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u(y_0)\hat{q}_u} + \frac{C_h}{2}\sum_{i=1}^{n_h} w_i k\left(\left\|\frac{y-x_i}{h}\right\|^2\right) \tag{6}$$

과 같고, 여기서

$$w_i = \sum_{u=1}^{m}\delta[b(x_i)-u]\sqrt{\frac{\hat{q}_u}{\hat{p}_u(y_0)}} \tag{7}$$

이다. 따라서, 식 (2)로 부터의 거리측도 $d(y)=\sqrt{1-\rho[\hat{p}(y),\hat{q}]}$ [12]를 최소화하는 문제는 식 (6)의 첫째 항이 상수이기 때문에 두 번째 항을 최대화하는 문제로 귀착되며 이는 커널함수 $k()$의 그림자 커널(shadow kernel) $g()$를 이용한 mean-shift에 의해 반복적으로 구해질 수 있다. 여기서 그림자 커널함수는

$$g(x) = -k'(x) \tag{8}$$

로 정의되며, 본 논문에서 이용된 가우시안 커널함수의 그림자 커널은 가우시안이다.

3.2 제안된 객체 추적방법

본 논문에서는 객체 추적에 신질에서 추정한 객체 움직임 플로우를 이용하였다. 제안된 방법에서 추적될 대상 객체는 바운딩박스에 의해 지정되며, MPEG으로 압축된 비디오 스트림에서는 픽셀 단위가 아니고 매크로 블록단위로 지정된다. 따라서, 3.1절에서 $\hat{p}(y), \hat{q}$는 지정된 객체의 매크로 블록들에서 움직임 플로우의 확률분포를 의미한다. 이들을 구성하는데 있어 반경 h이내에 있는 픽셀들 중 객체의 중심부분의 움직임 플로우들이 큰 확률 값을 가지게 될 수 있도록 커널함수는 exp(-x)를 사용하였다. 본 논문에서 제안된 객체 추적 알고리즘은 다

음과 같다.

Step 1: y_0 중심의 초기 객체지정 및 움직임 플로우 분포 $\hat{q}_u = \hat{p}_u(y_0)$ 계산

Step 2:

2-1) 다음 프래임에서 y_0 중심의 $\hat{p}_u(y_0)$ 및 $\rho[\hat{p}(y_0), \hat{q}]$ 계산

2-2) 식 (7)에 의한 가중치 $\{w_i\}_{i=1,\ldots,n_h}$ 계산

2-3) mean-shift에 의한 새로운 객체 중심 $\hat{y}_1$ 계산

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right)}$$

2-4) $\hat{y}_1$ 을 중심으로 $\rho[\hat{p}(y_1), \hat{q}]$ 계산

2-5) $\rho[\hat{p}(y_1), \hat{q}] < \rho[\hat{p}(y_0), \hat{q}]$ 을 만족하면 $\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_1 + \hat{y}_0)$ 을 반복

2-6) 만약 $\|\hat{y}_1 - \hat{y}_0\| < \varepsilon$ 추적완성, $\hat{q}_u = \hat{p}_u(y_1)$ 로 하고 Step 2 반복, 그렇지 않으면 $\hat{y}_0 \leftarrow \hat{y}_1$ Step 2-2부터 반복

이상의 알고리즘은 계속되는 프래임 움직임 플로우의 분포가 이전 프래임에서 객체의 움직임 플로우의 분포와 최대로 유사하도록 객체의 중심을 이동하는 방식으로 그림 9와 같이 표현할 수 있다. 일반적으로 I 프래임에는 움직임 벡터가 정의되지 않는다. 그러나, 전 절에서 추정된 객체의 움직임 플로우는 I 프래임에서도 보간된 움직임 프로우를 가질 수 있다. 따라서, 그림 9의 추적은 B 또는 P 프래임 뿐만 아니라 MPEG을 구성하는 모든 프래임에 대하여 동일한 방식으로 적용된다.

Ⅳ. 시스템 구성 및 실험 결과

앞의 2, 3절에서 제안된 객체 추적 알고리즘은 PC Windows 환경에서 Visual C++를 이용하여 구성되었다. 구성된 사용자 인터페이스에서는 초기 객체를 설정하기 위한 도구를 제공하고, 추적과정에서 매 프래임의 화면을 보여준다.

표 1은 4개의 GOP 당 15개의 프래임을 포함하는 MPEG-2 비디오 시퀀스의 제안된 방법을 이용한 추적의 개략적인 정확성을 보여주고 있다. 표에서 객체비율이란 매 GOP의 처음 I 프래임에서 바운딩 박스의 객체를 포함하는 매크로 블록과 객체 이외의 매크로 블록의 수를 표현하며 이 이들의 비율은 변화지 않아야 정확하다고 간주한 것이다. 실제로 객체가 강체가 아니기 때문에 이 비율은 추적이 성공적이라도 변할 수 있다. 이러한 점을 감안할 때 본 논문에서 제안된 객체추적은 약 4개의 GOP에서 성공적으로 시행되고 있다고 판단된다.

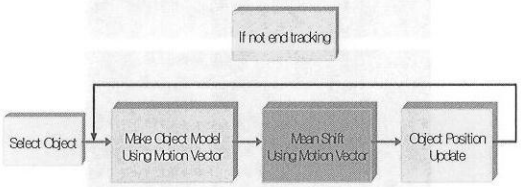


그림 10. 제안된 객체 추적 방법

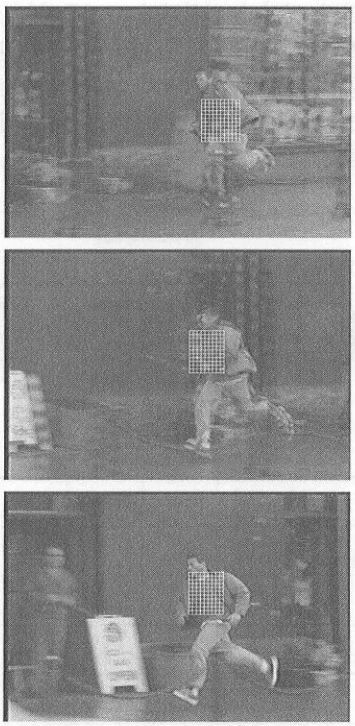
		1st I frame	2nd I Frame	3rd I Frame	4th I Frame	평균 오차
Test 1	객체비율 (객체/마스크)	10/10 (100 %)	10/10 (100 %)	7/10 (70 %)	7/10 (63 %)	
	오차비율		0 (0 %)	-3 (-30 %)	0 (0 %)	-1 %
Test 2	객체비율 (객체/마스크)	29/30 (96 %)	28/30 (93 %)	25/30 (83 %)		
	오차비율		-1 (-3 %)	-3 (-10 %)		-6.5 %
Test 3	객체비율 (객체/마스크)	10/18 (55 %)	11/18 (61 %)	10/18 (55 %)		
	오차비율		-1 (-10 %)	+1 (+9 %)		-0.5 %
Test 4	객체비율 (객체/마스크)	19/24 (79 %)	15/24 (62 %)	15/24 (62 %)	16/24 (66 %)	
	오차비율		-4 (-21 %)	0 (0 %)	+1 (6 %)	-5 %

오차비율: (오차 매크로 블록 개수) / (전I 프레임의 객체 매크로 블록 개수)
+, -: 객체 매크로 블록의 증감

표 1. 4개의의 테스트 비디오에 대한 실험 결과

그림 10은 표 1의 테스트 2의 비디오 스트림으로 카메라가 팬 동작을 지속하며 진행하는 사람을 추적하는 장면이다. 또한 그림 11은 객체를 추적하기 위해 보상된 x 방향의 팬양을 프래임별로 보여주고 있다. 그림 11에서 가로축은 프래임번호를 의미하며, 세로축은 픽셀수로 움직임 플로우의 보상될 양

을 의미한다. 이들 그림의 결과는 제안된 방법이 카메라 동작을 보상하기 때문에 카메라의 동작에 무관하게 객체를 추적할 수 있음을 보여주고 있다. 표 1에서 테스트 2의 오차는 달리는 사람이 강체가 아니고 MPEG의 움직임 벡터의 한계 때문에 얻어지는 결과로 분석할 수 있다. 이는 Favall 등이 제안한 방법에서는 카메라의 동작을 보상하지 않기 때문에 추적하기 어려울 것으로 예상된다.



90-Frame 103-Frame 113-Frame

그림 11. 카메라 움직임이 있는 경우의 추적

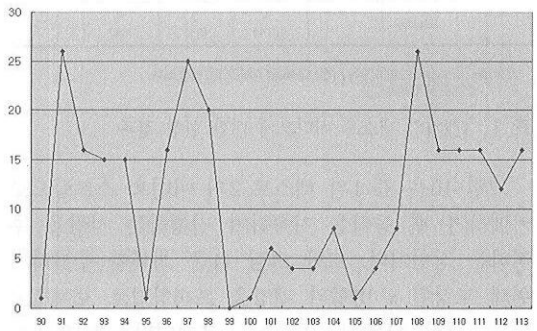


그림 13. 그림 10 비디오 시퀀스의 프레임별 x 방향의 팬량

V. 결 론

본 논문에서는 MPEG 비디오 스트림에서 복호화 과정없이 부호화에 필요한 매크로 블록당 움직임 벡터를 이용하여 움직이는 객체를 추적하는 방법을 제안하였다. 제안된 방법에서는 움직임 벡터로 부터 움직임 플로우를 추출하고, 카메라 동작에 따른 움직임을 보상하여 객체의 근사적인 움직임을 추출하는 방법을 제안하였으며 근사적인 객체의 움직임 플로우와 mean-shift 알고리즘을 이용하여 연속적으로 객체를 추적할 수 있다. 이러한 방법은 카메라의 동작을 보상하고 객체의 움직임만을 이용하기 때문에 카메라가 움직이는 경우에도 객체를 안정적으로 추적할 수 있는 장점도 가질 수 있다.

제안된 방법은 객체의 움직임 플로우를 압축영역에서 직접 얻음으로써 기존의 복호화된 영역에서의 추적에 비해 속도의 향상을 가져올 것으로 기대되며, 객체 추적에 따르는 탐색이 mean-shift 양에 의해 결정되므로 불필요한 탐색을 줄일 수 있다. 또한 여러 개의 객체를 동시에 추적하는 방식도 간단한 알고리즘의 변화로 실현 가능하다. 뿐만 아니라 제안된 객체 추적의 결과는 2차원 영상평면 내 근사적인 객체 움직임 해석도 가능할 수 있으며, 두 개 이상의 객체를 동시에 추적할 경우 이들 객체 상호간의 시간적인 변위량도 계산할 수 있을 것이다.
제안된 방법은 압축된 비디오에서의 객체 움직임 분석 및 해석된 움직임 정보의 저장 및 이를 이용하는 검색 등에 응용되리라고 기대된다.

참 고 문 헌

[1] Isaac Cohen, and Gerard Medioni, "Detection and Tracking of Objects in Airborne Video Imagery," CVPR 98 Workshop on Interpretation of Visual Motion, 1998.

[2] J. Meng, and S.-F. Chang, "Tools for Compressed domain Video Indexing and Editing,", SPIE Storage and Retrieval for Still Image and Video Databases, vol. 2670, pp.180-191, 1996.

[3] V. Kobla, and D. Doermann, "Compressed domain video indexing techniques using DCT and motion vector information in MPEG video," Proc. Of SPIE, vol. 3022,

pp.200-211, 1997.

[4] ISO-IEC 1-1/ISO-IEC 13812-2 International Standards, 1st Ed., 1996.

[5] Y. Nakajima, A. Yoneyama, H. Yanagihara, and M. Sugano, "Moving object detection from MPEG coded data," SPIE Visual Communications and Image Processing, vol. 3309, pp.988-996, 1998.

[6] R. Milanese, F.Deguillaume, A. Jacot-Descombes, "Efficient Segmentation and Camera Motion Indexing of Compressed Vide," Real-Time Imaging, vol. 5, No.4, pp. 231-241, Aug. 1999.

[7] Maurizio Pilu, "On using raw MPEG motion vectors to determine global camera motion," SPIE vol.3309, pp.449-459, 1998.

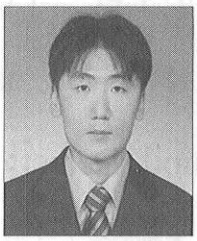
[8] Favalli, L., Mecocci, A., Moschetti, F.,"Object tracking for retrieval applications in MPEG-2," IEEE Transactions on Circuits and Systems for Video Technology, vol. 10, pp. 427-432, April 2000.

[9] Comaniciu, D.; Meer, P. "Mean shift: a robust approach toward feature space analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24 No. 5 , pp 603-619 May 2002.

[10] Yizong Cheng "Mean shift, mode seeking, and clustering," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17 No. 8 , pp. 790 -799, Aug. 1995.

[11] Comaniciu, D.; Ramesh, V.; Meer, P. "Real-time tracking of non-rigid objects using mean shift," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 13-15 June 2000.

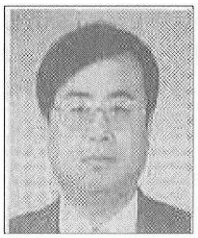
박 성 모(Sung-Mo Park)



2002년 2월 : 전북대학교 전자정보 공학부 공학사 졸업
2004년 2월 : 전북대학교 전자공학과 석사 졸업

<관심분야> 영상처리, 위성영상, 영상분할, MPEG-2

이 준 환(Joon-Whoan Lee) 정회원



1980년 2월 : 한양대학교 전자공학 공학사 졸업
1982년 2월 : KAIST 전기 및 전자 공학 석사 졸업
1990년 8월 : U. Of Missouri(미국 미주리주 콜럼비아) ECE 공학박사 졸업
1990년 10월 ~ 현재 : 전북대학교 전자정보 공학부 근무

<관심분야> 영상처리, 영상분할, MPEG-2, MPEG-7