

MPEG-2 압축 영역에서 움직이는 객체의 추적 및 해석

문수정[†]·유원영[†]·김준철^{††}·이준환^{†††}

요 약

본 논문에서는 MPEG-2비디오 스트림에서 복호화 과정없이 압축비디오에서 직접 얻을 수 있는 정보들을 활용하여 움직이는 객체를 추적하고 해석하는 방법을 제안한다. 제안된 방법에서는 먼저 MPEG-2의 움직임 벡터로부터 근사적으로 움직임 플로우(motion flow)를 구성하고, 전역적인 움직임 플로우로부터 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 정의함으로써 시동된다. 이후의 객체의 추적은 카메라 움직임이 보정된 객체의 움직임 플로우를 한 GOP(Group of Pictures)단위로 면적 기여도에 따라 누적하여 추적하였다. 또한 추적오차의 누적을 막기 위해 매 GOP마다 DCT(Discrete Cosine Transform) 정보를 이용하여 초기 바운딩 박스와 매칭을 통해 객체의 영역을 재설정하였다. 제안된 방법은 압축된 비디오 스트림에서 직접 정보를 얻음으로써 계산속도의 향상을 기할 수 있으나, 압축된 MPEG-2 비디오에서 얻을 수 있는 정보들이 최대 블록 단위이므로 객체의 정의도 블록단위 이상의 객체로 제한되며, 이용할 수 있는 정보가 제한되어 있기 때문에 정확한 객체추적보다는 근사적인 객체추적에 적합하다.

Tracking and Interpretation of Moving Object in MPEG-2 Compressed Domain

Su-Jeong Mun[†] · Woon-Young Ryu[†] · Joon-Cheol Kim^{††} · Joonwhoan Lee^{†††}

ABSTRACT

This paper proposes a method to trace and interpret a moving object based on the information which can be directly obtained from MPEG-2 compressed video stream without decoding process. In the proposed method, the motion flow is constructed from the motion vectors included in compressed video. We calculate the amount of pan, tilt, and 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. Initially, a moving object to be traced is designated by user via bounding box. After then automatic tracking is performed based on the accumulated motion flows according to the area contributions. Also, in order to reduce the cumulative tracking error, the object area is reshaped in the first I-frame of a GOP by matching the DCT coefficients. The proposed method can improve the computation speed because the information can be directly obtained from the MPEG-2 compressed video, but the object boundary is limited by macro-blocks rather than pixels. Also, the proposed method is proper for approximate object tracking rather than accurate tracing of an object because of limited information available in the compressed video data.

키워드: 압축비디오(Compressed Video), 객체 추적(Object Tracking), 일반화된 Hough 변환(Generalized Hough Transfer), 움직임벡터(Motion Vector), 움직임 플로우(Motion Flow)

1. 서 론

컴퓨터의 발달과 인터넷의 확산으로 멀티미디어 콘텐츠의 보급이 급속히 확대되고 있다. 이들 콘텐츠에는 원격러 화상회의, 감시시스템, 주문형 비디오(VOD), 주문형 뉴스(NOD), 디지털 편집 시스템 등 동영상의 포함되어 있다. 그러나 이와 같은 동영상 서비스를 한정된 대역폭을 통해 저장하거나 전송하기 위해서는 동영상이 가지는 방대한 양

의 데이터를 MPEG과 같은 방식으로 압축하는 것이 필수적이다.

디지털 비디오의 압축 표준은 디지털 방송콘텐츠를 대상으로한 MPEG-2와 낮은 전송율에 초점을 맞춘 객체지향 부호화 방식인 MPEG-4, 비디오 데이터의 저장 및 검색용도의 MPEG-7 등으로 발전되어 왔다. 현재 상용화되어 가장 보편적으로 사용하는 MPEG 콘텐츠는 MPEG-2이며 본 논문의 범위는 MPEG-2에 한정한다. 즉 현재 연구가 진행 중인 객체지향 부호화 방식인 MPEG-4 및 움직이는 객체의 표현자(descriptor)를 대상으로한 MPEG-7 등에서의 객체 분할과 추적은 본 논문의 범위에 포함되지 않는다.

본 논문에서는 이와 같이 보편화되어 사용되고 있는 MP

* 본 연구는 정보통신부 지원 대학 ITRC 사업에 의해 부분적으로 지원되어음.
[†] 정 회 원 : 한국전자통신연구원 연구원
^{††} 정 회 원 : 서남대학교 전자공학과 교수
^{†††} 정 회 원 : 전북대학교 전자정보공학부 교수
 논문접수 : 2003년 7월 4일, 심사완료 : 2003년 12월 5일

EG-2로 압축된 데이터에 포함된 정보를 효율적으로 이용하여, 검색 및 그와 관련된 데이터 베이스의 구축에 활용되는 움직임은 객체의 추적 및 해석을 다루고자 한다. 이러한 목적을 위해서 기존의 방법들에서는 압축된 동영상 데이터를 복호화 한 후 시공간 영역에서 처리하였는데 다양한 정보를 비디오로부터 얻을 수 있다는 융통성이 있는 반면에 속도저하는 물론, 부호화 과정에서 발생하는 유용한 정보 즉 움직임 정보나 블록의 질감 정보 등의 간과하고 다시 계산될 수 있다는 단점을 내포하고 있다.

이러한 단점을 극복하기 위하여 압축영역에서의 압축에 수반된 다양한 정보를 이용하고자 시도하였는데, S-F Chang 등은 DCT(Discrete Cosine Transform) 영역에서 질감을 분석하는 여러 방법을 제안하였다[1]. 또한 V. Kobla는 MPEG의 움직임 벡터를 이용하여 압축영역에서 움직임 플로우 개념을 확립하였으며[2], A. Yoneyama와 Y. Nakajima는 압축영역에서 움직이는 객체를 정의하고 해석하였으며, Favalli 등은 객체 추적에 움직임 벡터를 이용할 수 있음을 보였다 [3, 8, 9].

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 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 제거함으로써 해결할 수 있다. 또한 전자의 문제점의 경우 카메라의 움직임이 전역적인데 반하여 객체의 움직임은 국부적인 사실을 이용하여 단순한 카메라 모델을 정의하고 부수되는 파라미터를 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 보정한 후 매크로 블록 당 국부적인 객체의 움직임을 계산하고, 이를 이용하여 사용자에게 의해 미리 정의된 바운딩 박스내의 객체를 GOP 단위로 추적하는 방법을 제안하였다. 제안된 방법에서는 한 GOP 내의 B, 또는 P 프레임에서는 계산된 객체의 누적 움직임 플로우를 이용하고, 새로운 GOP의 I 프레임에서는 추적 오차의 누적을 막기 위해 DCT계수들을 이용하여 추적객체를 재설정하였다. 또한 추적되는 객체의 움직임의 간단한 2차원 평면상의 해석이 가능함을 실험을 통해 입증하였다. 본 논문의 제안된 방법은 Favalli 등의 방법과 유사하나[9], 카메라 움직임 보상을 수행한 후에 객체의 추적이 진행되기 때문에 카메라의 움직임과는 무관하게 객체의 추적이 가능하다.

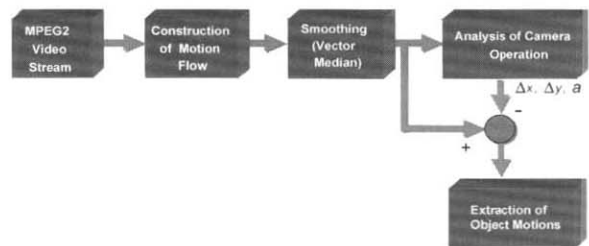
본 논문의 이러한 접근방법은 객체지향 부호화 방식인

MPEG-4는 내용기반 비디오 서비스에 필요한 유용한 정보를 프레임 기반인 MPEG-1 또는 MPEG-2 보다 많이 포함하고 있기 때문에 유리하다. 그러나, 일반 영상의 분할 문제가 완벽하게 해결되지 않았기 때문에 구형(rectangular)영상의 경우 VOP를 프레임 또는 필드로 규정하고 있다. 따라서, 본 논문에서는 현재 보편적으로 이용되고 있는 MPEG-2를 대상으로 한다[10, 11].

본 논문의 2장에서는 MPEG-2 비디오 스트림으로부터 객체의 움직임을 추출하는 방법을 기술하였으며, 3장에서는 움직이는 객체의 추적방법에 대해서 설명하였고, 4장에서는 구현된 시스템 및 실험결과에 대해서 논하였다. 마지막으로 5장에서는 본 논문의 결론 및 확장방법에 대해서 논하였다.

2. 객체의 움직임 추출

본 논문에서 제안된 객체 추적알고리즘은 압축된 MPEG-2 비디오에 담겨있는 정보를 복호화 과정없이 직접 매크로 블록당 움직임을 추출한다. (그림 1)은 MPEG-2 비디오 스트림에서 객체의 움직임 플로우를 추출하는 과정을 보여주고 있다. 그림에서 알 수 있듯이 객체의 움직임을 추출하는 과정은 MPEG-2 비디오 스트림에서 움직임 플로우를 추정하고, 추정된 움직임 플로우를 시공간 벡터 미디언 필터를 이용하여 평활화하며, 카메라 동작을 보상하여 얻어진다.



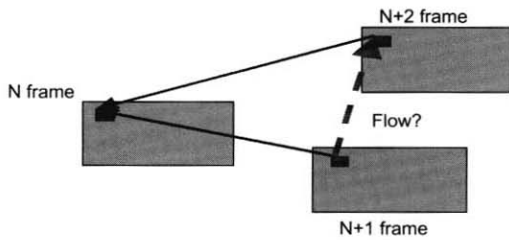
(그림 1) 객체 움직임의 추출

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. 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)는 매크로 블록단위의 움직임 벡터로부터 움직임 플로우 예측의 필요성을 보여주고 있다. 그림에서 N 프레

입은 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.

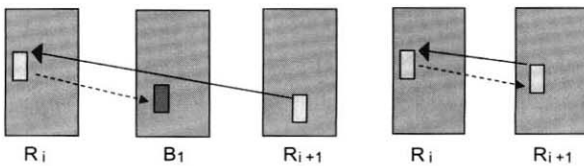


(그림 2) 매크로 블록단위의 움직임 플로우 예측

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

- (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 $B(t) + P(t+1)$
- (d) $I(t) + B(t+1)$ 또는 $P(t) + B(t+1)$

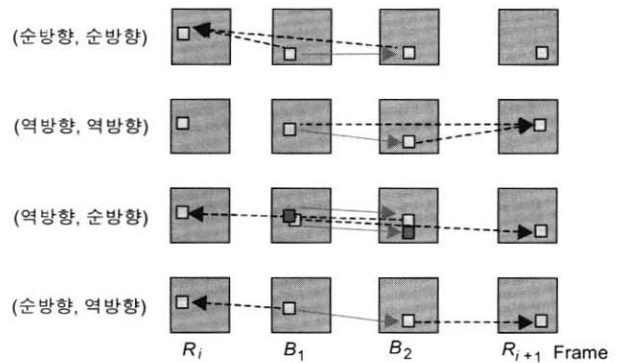
식 (1-a)에서의 경우는 (그림 3)에 표현된 바와 같다. (그림 3)의 첫 번째 경우는 순방향 예측된 움직임 벡터의 반대 방향을 움직임 플로우는 가정할 수 있는 경우를 의미하고, 두 번째 경우는 두 참조 프레임 사이에 B 프레임이 삽입된 경우로 움직임 벡터의 크기를 삽입된 프레임수+1로 나누고 방향을 반대로 표현하는 경우이다.



(그림 3) 참조 프레임간의 움직임 플로우 추정

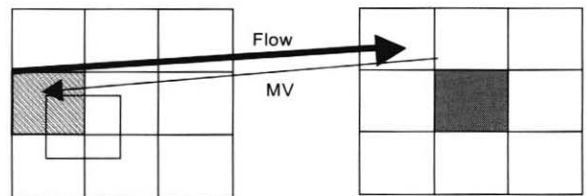
(그림 4)는 식 (1-b)의 경우로 움직임 벡터의 종류에 따라 (순방향, 순방향), (역방향, 순방향), (역방향, 역방향)인 경우에 각 움직임 벡터의 차이에 의해 움직임 플로우는 구성할 수 있다. 그러나, (그림 4)의 마지막에 표현된 (순방향, 역방향)인 경우에는 움직임 플로우는 구성할 수 없다.

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



(그림 4) B 프레임간의 움직임 플로우 추정

또한 식 (1)에서 예시한 움직임 플로우는 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 5)의 매크로 블록이 해당 움직임 벡터의 역방향의 움직임 플로우는 갖는다고 가정하였다.

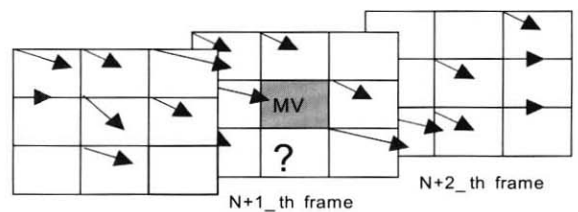


(그림 5) 움직임 벡터와 역 방향인 움직임 플로우는 구성

2.2 움직임 플로우는 보간 및 평활화

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

이러한 시공간적인 잡음을 제거하고, 움직임 플로우는 정의되지 않는 매크로 블록을 보간하기 위해 (그림 6)과 같이 $3 \times 3 \times 3$ 크기의 윈도우를 이용하여 시공간적인 메디안 필터링을 수행하였다. 이러한 메디안 필터의 결과는 시공간적인 움직임의 불연속을 보존하면서 불규칙한 잡음 형태의 움직임 플로우는 평활화 할 수 있다.



(그림 6) 벡터 메디안에 의한 시공간 필터링

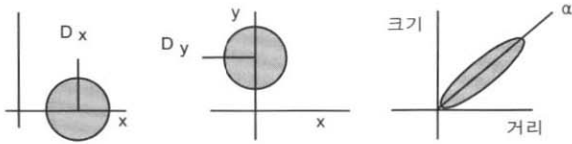
(그림 6)의 필터링 대상은 움직임 플로우에 해당하는 벡터정보로서 매디안 벡터는 해당 윈도우 내의 움직임 플로우 벡터간의 유클리디안 거리의 합을 최소로 하는 벡터에 해당한다. 이러한 총체화 정렬(aggregate ordering)방식을 이용한 벡터정보의 순서통계(order statistics) 필터링은 벡터의 불연속을 보존하며 가우시안 형태의 잡음을 효과적으로 제거할 수 있으며, 움직임 플로우가 정의되지 않은 I, B, P 프레임의 매크로 블록들에 대해서 시공간적으로 인접한 움직임 플로우로부터 보간된 움직임플로우로 채워질 수 있다.

2.3 카메라 동작의 추정

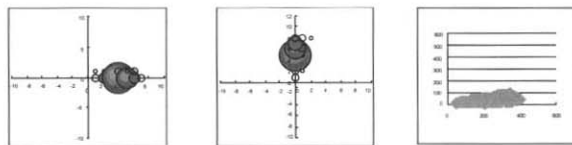
카메라의 움직임은 아파인(affine) 모델과 원근투영(perspective-projection) 모델 등을 이용하여 근사화 할 수 있다. 그러나 이러한 접근은 압축정보에서 구할 수 없는 카메라의 초점거리를 사용해야 하므로 본 논문에서는 가장 단순한 팬-틸트-줌(pan-tilt-zoom) 모델을 사용했다[6, 7].

이러한 모델에서 R.Milanese등은 프레임의 대표 값인 Δx , Δy 를 구하기 위해서 사용하였는데, 이들은 최소자승(least-meansquare) 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 Δx , Δy 는 전체 매크로블록의 50% 이상이 움직임 벡터를 가지면서 그 중 60%이상이 한 그룹으로 포함될 때, 그 그룹의 벡터 매디안으로 대표되는 움직임 벡터성분을 Δx , Δy 로 정의하였다. 또한 줌의 중심좌표를 중점으로 가진 원을 기준으로 밖으로 향하는 성분크기를 양의 값, 안쪽으로 향하는 성분의 크기를 음의 값을 가진 크기와 중심부 터의 거리로써 Hough 공간을 정의하고 가장 많은 분포를 대표하는 α 기울기를 α 로 정의하였다.



(a) 이상적인 매개변수 분포도

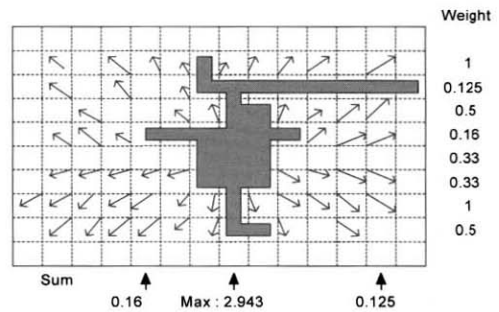


(b) 실제 영상에서의 매개변수 분포도
(그림 7) 팬 틸트 줌의 매개변수 분포도

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

에서 구한 매개변수의 분포도이다.

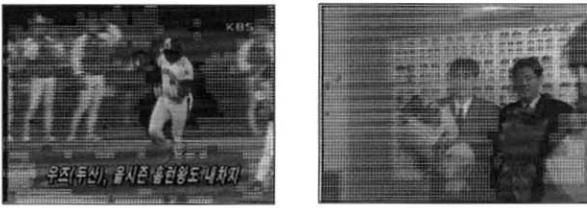
카메라 동작을 정의하기위한 x, y의 매개변수는 매 프레임마다 계산 되어지며, 연속적인 카메라 움직임을 추정할 수 있다. 또한 값을 계산하기위한 중심좌표는 매개변수 값을 결정하는데 중요한 요인이 되는데 추정에러를 줄이고자 각 프레임마다 가로와 세로방향으로 움직임 플로우의 y와 x성분이 없는 매크로 블록들에게 가중치를 주어, 누적하여 중심좌표를 구하는 방법을 취하였다. (그림 8)은 확대(zoom-in)가 발생할 때 각 매크로 블록들의 움직임 벡터의 상태와 가중치를 표현하고, 가장 많이 누적된 중심의 x 좌표를 찾는 그림의 예이다. 그림에서 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 x 성분 구하는 예

2.4 카메라 동작을 배제한 움직임 추출

MPEG-2 비디오의 움직임 벡터는 카메라 동작과 객체의 움직임이 혼재되어 나타난다. 즉 카메라의 동작에 의한 움직임이 프레임 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 Δx , Δy 값의 움직임을 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 α 값(기울기)에 따라 움직임 플로우의 값을 빼면 카메라 동작을 제거한 순수한 움직임은 객체의 움직임을 추정할 수 있다. (그림 9)는 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

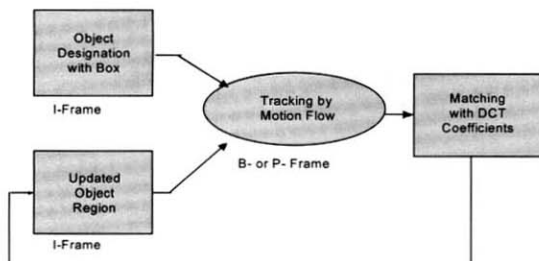
3. 제안된 객체 추적 알고리즘

전 절에서 추정된 객체의 움직임 플로우를 이용한 제안된 객체추적 과정은 한 GOP단위로 이루어진다. GOP 단위로 구성된 MPEG-2 비디오 스트림에서 객체의 움직임을 추적하기 위해서 활용될 수 있는 정보는 I 프레임에서는 블록 또는 매크로 블록단위의 DCT 계수 값들과 B 또는 P 프레임에서는 매크로 블록 단위의 움직임 플로우이다.

제안된 추적 방법에서 추적 대상 객체는 최초의 GOP에서 바운딩 박스에 의해 지정된다. MPEG-2 에서 추정된 움직임 플로우와 DCT 계수들은 매크로 블록 단위로 계산되기 때문에 바운딩 박스는 기존의 비압축 영역에서의 객체 지정과는 다르게 매크로 블록에서 단위로 지정된다.

I 프레임의 DCT계수 정보만을 이용하여 추적한다면 새로운 I 프레임을 만날 때까지 GOP를 구성하는 프레임 수만큼을 건너가서 새로운 GOP에서 해당 객체를 찾아야 한다. 만약 GOP가 15개의 프레임으로 구성된다면 15개의 프레임을 지나고 난 후의 새로운 I 프레임에서 객체를 찾아야 하기 때문에 객체의 움직임이 빠를 경우에는 해당 객체의 탐색 범위가 넓어지고 탐색에 소요되는 시간이 길어진다.

제안된 방식에서 지정된 객체는 연속되는 B, P 프레임들에서 객체 움직임 플로우를 이용하여 추적을 진행하며 새로운 GOP의 시작 I 프레임에서 탐색은 그때까지의 추적 결과를 이용하여 탐색범위를 제한한다. 제안된 추적 알고리즘의 개요는 (그림 10)과 같다.



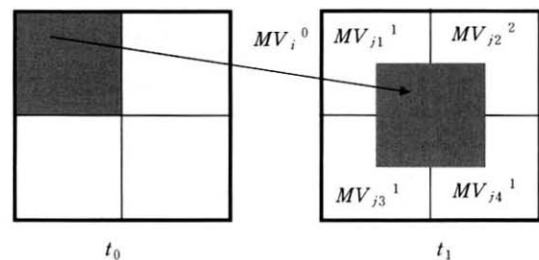
(그림 10) 객체 추적 알고리즘의 개요

3.1 면적기여도를 이용한 누적된 움직임 및 객체추적

전 절에서 구성한 움직임 플로우는 매크로 블록 단위로 이전 프레임의 한 매크로 블록이 현재의 프레임에서 얼마

나 이동되었는가를 표현한다. 따라서 초기의 어떤 매크로 블록이 두개 이상의 연속되는 프레임을 지나면서 어떻게 움직였는가를 알아내기 위해서는 특별한 고려가 필요하다. 본 논문에서는 B 또는 P 프레임에서 면적 기여도를 이용하여 초기의 I 프레임의 매크로 블록이 어떠한 움직임을 가지고 이동하는가를 추정하였다.

(그림 11)은 면적기여도에 따른 움직임 플로우를 누적하는 방법을 보여주고 있다. 초기에 지정된 객체내의 한 매크로 블록이 움직임 플로우에 의해 다음 프레임에서 변위가 일어났다고 가정하자. 변위에 의해 다음 프레임의 일반적으로 4개의 매크로 블록들과 중첩되어 나타난다. 이 경우 초기의 매크로 블록은 두개의 연속된 프레임을 거치면서 얼마의 변위 총합을 가지게 될 것인가는 중첩된 4개의 매크로 블록들의 움직임 플로우를 중첩된 면적에 따라 스케일링하여 초기 변위량에 더할 수 있다. 그림에서는 t_0 프레임에서 음영된 매크로 블록이 t_1 프레임에서 4개의 매크로 블록에 겹쳐져 나타나는 과정을 보여 주고 있다.



(그림 11) 누적 움직임 플로우의 계산

일반적으로 누적된 움직임 플로우는

$$S_{MV} = S_{MV} + \frac{A_1 MV_1 + A_2 MV_2 + A_3 MV_3 + A_4 MV_4}{A_1 + A_2 + A_3 + A_4} \quad (2)$$

와 같이 구해진다. 여기서 S_{MV} 는 총체화된 움직임을 표현하고, $MV_1 \sim MV_4$ 는 연속된 프레임에서 중첩된 매크로 블록의 움직임 플로우를 표현하며, $A_1 \sim A_4$ 는 중첩된 부분의 면적을 표현한다.

추적이 진행되면서 면적 기여도별로 누적 시킨 움직임 플로우의 누적된 값이 8 화소 이상이면 주변의 매크로 블록도 객체가 반 이상 겹쳐져 있는 형태를 가짐을 의미한다. 제안된 방법에서는 이 경우 그 주변의 매크로 블록도 객체의 영역으로 간주하였다[9]. 또 12 화소 이상 움직였을 경우는 해당 매크로 블록에서 객체가 3/4 이상 빠져간 형태이므로 해당 매크로 블록을 객체의 영역에서 제거하였다. 이러한 방식으로 추적이 진행되어 한 GOP가 지나면 새로운 I 프레임에서 추정된 객체의 영역을 중심으로 후보 영역들을 설정한다. 움직임 플로우를 이용하여 B 또는 P 프레임에서의 추적과정을 정리하면 다음과 같다.

```

Tracking by Motion Flow in B- or P- frame
For each succeeding B- or P-frame
  For each macro block occupied by object boundary in the
  previous frame
    Calculate accumulated motion flow S_MV
    If S_MV < 8
      Select the same macro block
      Continue S_MV
    If 8 <= S_MV <= 12
      Expand a macro block for object
      Continue S_MV and Start S_MV for expanded MB
    If S_MV >12
      Replace the new macro block
      Stop S_MV and Start S_MV for replaced MB
  End for
End for
    
```

3.2 객체 재설정 및 객체 움직임 해석

전 절에서 추적된 객체(실체는 객체를 포함하는 매크로 블록)는 한 GOP가 지난후 새로운 GOP의 시작 I 프레임에서 객체의 중심에 대한 후보 매크로 블록이 될 수 있다. 따라서, 새로운 I 프레임에서의 객체의 재지정은 현재까지의 추적된 후보 매크로 블록들을 중심으로 한 바운딩박스과 이전의 GOP의 시작 I 프레임에서의 바운딩 박스와 매칭을 통하여 재설정 된다. 즉 후보 매크로 블록 각각을 객체 영역의 중심으로 한 객체 영역 바운딩 박스를 취하였을 때의 각 DCT의 DC계수의 값과 전 프레임에서 찾은 객체 영역의 DCT값을 비교하여 오차가 가장 적은 매크로 블록을 객체의 중심으로 하여 새로운 객체 영역을 설정한다.

식 (3)은 현재 까지 추적된 매크로 블록들을 중심 (b_i, b_j) 으로 한 바운딩 박스 내의 블록 단위의 Y 성분의 DCT DC계수 값과 매크로 블록 단위의 Cb, Cr 성분의 DCT DC 성분 값과 이전 GOP에서 바운딩 박스의 DC 성분값의 오차를 계산하는 수식이다. 또한 식 (2)에서 계산된 현재까지 추적된 모든 후보 매크로 블록들에 오차로부터 식 (4)와 같이 최소의 오차를 갖는 매크로 블록을 중심으로 같은 크기의 바운딩 박스로 객체 영역을 재설정하였다.

$$\begin{aligned}
 Error(b_i, b_j) = & \sum_{\substack{\text{blocks in} \\ \text{bounding box}}} |DC_Y(i - y_c^{old}, j - x_c^{old}) \\ & - DC_Y(i - b_i, j - b_j)| + \\ & \sum_{\substack{\text{Macro Blocks in} \\ \text{bounding box}}} |DC_Cb(i - y_c^{old}, j - x_c^{old}) \\ & - DC_Cb(i - b_i, j - b_j)| + \\ & \sum_{\substack{\text{Macro Blocks in} \\ \text{bounding box}}} |DC_Cr(i - y_c^{old}, j - x_c^{old}) \\ & - DC_Cr(i - b_i, j - b_j)| + \\ & (y_c^{new}, x_c^{new}) = \text{argmin } Error(b_i, b_j) \quad (3)
 \end{aligned}$$

제안된 방법에서 객체 움직임의 해석은 매 GOP 단위로

수행되며, 해석될 움직임의 종류는 4가지로 (위, 아래, 오른쪽, 왼쪽)로 표현된다. 새로운 GOP의 시작 I 프레임의 갱신된 바운딩 박스의 중심값 (x_c^{new}, y_c^{new}) 과 이전 GOP의 시작 I 프레임에서의 바운딩 박스의 중심값 (y_c^{new}, x_c^{new}) 의 차이는 근사적으로 2차원 평면상의 객체의 움직임 해석을 제공한다.

즉 $\Delta x = (x_c^{new}, x_c^{new}), \Delta y = (y_c^{new}, y_c^{new})$ 라 할 경우

$|\Delta x| \geq |\Delta y|$ 이고 $\Delta x \geq 0$ 이면 '오른쪽'

$|\Delta x| \geq |\Delta y|$ 이고 $\Delta x < 0$ 이면 '왼쪽'

$|\Delta x| \geq |\Delta y|$ 이고 $\Delta y \geq 0$ 이면 '위른쪽'

$|\Delta x| \geq |\Delta y|$ 이고 $\Delta y < 0$ 이면 '아래쪽'

객체가 진행한다고 해석하였다.

4. 시스템구성 및 실험 결과

앞의 2장, 3장에서 제안된 객체 추적 알고리즘은 PC Windows 환경에서 Visual C++를 이용하여 구성되었다. 구성된 사용자 인터페이스에서는 초기 객체를 설정하기 위한 도구를 제공하고, 추적과정에서 매프레임의 화면을 보여주며, 객체의 움직임을 해석하여 표시할 수 있다.

<표 1>은 4개의 MPEG-2 비디오 시퀀스의 제안된 방법을 이용한 추적의 개략적인 정확성을 보여주고 있다. 실험에 사용된 비디오는 줌과 객체의 움직임이 동시에 나타나는 경우가 드물고, 본 논문에서 제안한 추적방식이 줌의 경우 적합하지 않기 때문에 줌에 의한 카메라 동작을 포함하는 시퀀스는 제외하였다. 표에서 객체비율이란 I 프레임에서 바운딩 박스의 객체를 포함하는 매크로 블록과 객체 이외의 매크로 블록의 수를 표현하며 이 이들의 비율은 새로운 GOP의 시작 I 프레임에서 객체가 재설정될 때 변화하지 않아야 정확하다고 간주한 것이다. 실제로 객체가 강체가 아니기 때문에 이 비율은 추적이 성공적이라도 변할수 있다. 표의 테스트 4는 비교적 크게 포착된 달리는 사람으로 이러한 사실을 반영하고 있으며, 테스트 3의 결과는 자전거를 탄 장면으로 자전거의 바퀴부분에서 배경이 겹쳐 나타나기 때문에 추적오차가 발생하는 것으로 나타나고 있다. 이러한 점을 감안할 때 본 논문에서 제안된 객체추적은 성공적으로 수행되고 있다고 판단된다.

또한 (그림 12)는 테스트 3의 추적결과를 보여주고 있다. 그림에서 알수 있드시 GOP 내에서 B 또는 P 프레임에서의 추적은 설정된 바운딩 박스를 변형시켜 나가고 있음을 알 수 있다. 즉 누적된 움직임 플로우와 매크로 블록의 확장 및 교체에 의해 직사각형 형태의 바운딩 박스는 변형되

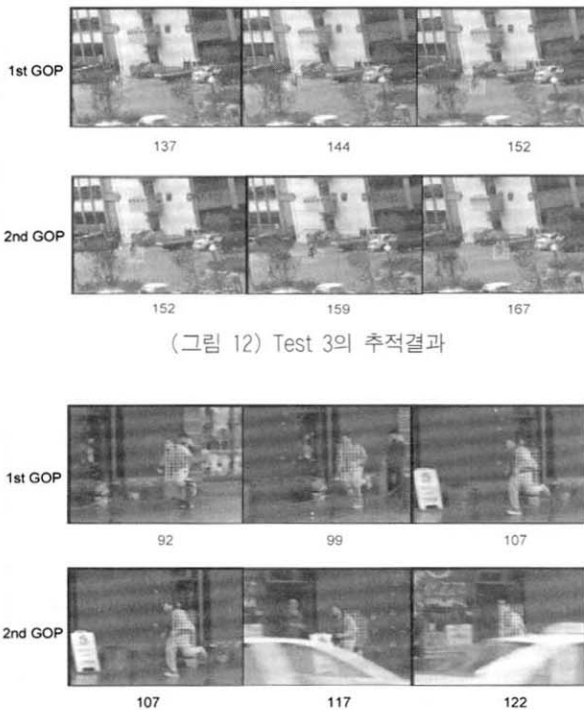
거나 분리 되어 존재할 수 있다. 이는 추정된 움직임 플로우가 움직임의 전면과 후면에만 발생하는 움직임 벡터를 근간으로 하기 때문에 발생하는 결과로 분석할 수 있다.

(그림 13)은 카메라가 팬 동작을 지속하며 진행되는 사람을 추적하는 장면이다. 본 논문의 제안된 방법은 카메라 동작을 보상하기 때문에 카메라의 동작에 무관하게 객체를 추적할 수 있음을 보여주고 있다. 이는 Favall 등이 제안한 방법에서는 카메라의 동작을 보상하지 않기 때문에 추적하기 어려울 것으로 예상된다.

<표 1> 4개의 테스트 스트림의 결과

		1st frame	2nd frame	3rd frame	4th frame	평균오차
Test 1	객체비율 (객체/마스킹)	11/16 (69%)	9/16 (56%)	10/16 (63%)	10/16 (63%)	
	오차 비율		-2 (-13%)	+1 (+7%)	0 (0%)	-2%
Test 2	객체비율 (객체/마스킹)	45/48 (94%)	44/48 (92%)	39/48 (81%)		
	오차 비율		-1 (-2%)	-5 (-11%)		-6.5%
Test 3	객체비율 (객체/마스킹)	52/54 (96%)	47/54 (87%)	40/54 (74%)	40 / 54 (74%)	
	오차 비율		5 (-9%)	-7 (-13%)		-11%
Test 4	객체비율 (객체/마스킹)	13/20 (65%)	11/20 (55%)	8/20 (40%)		
	오차 비율		-2 (-10%)	-3 (-15%)		-12.5%

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



(그림 12) Test 3의 추적결과

(그림 13) Test 4의 추적 결과



(그림 14) 사용자 인터페이스 및 움직임 해석

또한 움직임 해석의 경우 실험된 모든 경우에 정확한 해석을 부여하였으며 (그림 14)는 테스트 4의 움직임 해석의 결과와 구성된 사용자 인터페이스를 보여주고 있다.

5. 결 론

본 논문에서는 MPEG-2 비디오 스트림에서 복호화 과정 없이 부호화에 필요한 매크로 블록당 움직임 벡터와 DCT 계수 정보들을 이용하여 움직이는 객체를 추적하는 방법을 제안하였다. 제안된 방법에서는 움직임 벡터를 이용하여 움직임 플로우를 추출하고, 카메라 동작에 따른 움직임을 보상하여 객체의 근사적인 움직임을 추출하는 방법을 제안하였으며 근사적인 객체의 움직임을 이용하여 한 GOP내의 B 또는 P 프레임에서의 추적을 진행하고 새로운 GOP의 시작 I 프레임에서 객체를 재설정함으로써 객체추적에 따르는 오차를 누적시키지 않을 수 있으며, 객체 재설정에 따르는 탐색 범위를 제한할 수 있다. 또한 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 디코딩된 영역에서 하던 추적에 비해 속도의 향상을 가져올 수 있으며, 여러 개의 객체를 동시에 추적하는 방식도 간단한 알고리즘의 변화로 실현가능하고, 이들 객체들의 2차원 영상평면내에서 상대적인 움직임 해석도 가능하다.

제안된 방법은 현재 연구되고 있는 MPEG-4 또는 MPEG-7등의 객체 분할과 추적기술[12]이 충분히 유용한 단계까지는 가장 많이 사용되고 있는 MPEG-2 등의 압축된 비디오에서의 객체 움직임 분석 및 해석된 움직임 정보의 저장 및 이를 이용하는 검색등에 응용되리라고 기대된다.

참 고 문 헌

[1] 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.

[2] V. Kobla and D. Doermann "Compressed domain video indexing techniques using DCT and motion vector information in MPEG video," *In Proc. Of SPIE*, Vol.3022, pp. 200-211, 1997.

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

[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 Video," *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 p.449-459, 1998.

[8] 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.

[9] 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.

[10] 고성제, 김종욱, *MPEG-4의 세계*, 영풍문고, 2000.

[11] 김대회, 호요성, "MPEG-4 표준을 위한 동영상 객체분할 기술", *Telecommunication Review*, 제13권 제2호, Apr., 2003.

[12] Cigdem Eroglu Erden, A. Murat Tekalp, and Bulent Sankur, "Video Object Tracking with Feedback of Performance Measure," *Transactions on Circuits and Systems for Video Technology*, Vol.13, No.4, pp.310-324, April, 2003.



문 수 정

e-mail : sjmun@etri.re.kr

1996년~2000년 전북대학교 전자공학과
학사학위

2000년~2002년 전북대학교 대학원 석사
학위

2002년~현재 한국전자통신연구원 재직

관심분야 : 멀티미디어 시스템, 영상처리, 멀티미디어 스트리밍
솔루션 개발



유 원 영

e-mail : zero2@etri.re.kr

1996년~1998년 전북대학교 대학원 석사 학위

1989년~1996년 전북대학교 전자공학과
학사학위

1998년~2003년 전북대학교 대학원 박사 학위

2001년~현재 한국전자통신연구원 재직

관심분야 : 비디오 분석 및 편집, 영상처리, 디지털 워터마킹,
디지털 팡거프린팅



김 준 철

e-mail : jkkim@tiger.seonam.ac.kr

1986년 전북대학교 전자공학과(석사)

1988년 전북대학교 대학원 전자공학과
(공학석사)

1988년~1991년 LG 산전 연구소 연구원

1991년~1995년 전북대학교 대학원 전자

공학과(공학박사)

1993년~현재 서남대학교 전기전자통신공학부 조교수

관심분야 : 영상처리 및 컴퓨터비전



이 준 환

e-mail : chlee@moak.chonbuk.ac.kr

1980년 한양대학교 전자공학과(학사)

1982년 한국과학기술원 전자공학과
(공학석사)

1982년~1985년 전북대학교 전자공학과 조교

1985년~1987년 전북대학교 전자공학과

전임강사

1990년 미주리대 전산학과(공학박사)

1990년~1994년 전북대학교 전자공학과 조교수

1994년~현재 전북대학교 전자공학과 부교수

관심분야 : 영상처리 및 분석, 인공지능