

MPEG 압축 영역에서의 움직이는 객체 추적 및 해석

문수정, 이준환, 박동선
전북대학교 전자공학과
전북 전주시 덕진구 덕진동 1가 664-14

A Study on The Tracking and Analysis of Moving Object in MPEG Compressed domain

Su-Jung Mun , Joon-Whoan Lee and Dong-Sun Park
Chonbuk University of Division of Electronic and Information Engineering
E-mail: sjmun@ailab.chonbuk.ac.kr

요약

본 논문에서는 MPEG2비디오 스트림에서 직접 얻을 수 있는 정보들을 활용하여 카메라의 움직임을 추정하여 이를 기반으로 하여 움직이는 객체를 추정하고자 한다. 이를 위해, 먼저 MPEG2의 움직임 벡터는 압축의 효율성 때문에 움직임의 예측이 순서적이지 못한데, 예측 프레임들의 속성을 이용하여 이를 광플로우(Optical Flow)를 갖는 움직임 벡터(Motion Vector)로 변환하였다. 그리고 이러한 벡터들을 이용하여 카메라의 기본적인 움직임인 팬(Pan), 틸트(Tilt), 줌(Zoom)등을 정의하였다. 이를 위하여 팬, 틸트-줌 카메라 모델의 매개변수와 같은 의미의 Δx , Δy , α 값을 정의하고자 움직임 벡터 성분의 Hough변환을 이용하여 Δx , Δy , α 값들을 구하였다. 또 한 이러한 카메라 움직임(Camera Operation)은 시간적으로 연속적으로 발생하는 특징을 이용하여 각 프레임마다 구한 카메라의 움직임을 보정하였다. 마지막으로 움직이는 객체의 추정은 우선 사용자가 원하는 객체를 바운딩박스 형태로 정의한 후 카메라 움직임이 보정된 객체의 움직임 벡터를 한 GOP(Group of Pictures) 단위로 면적 기여도에 따라 누적하여 객체를 추적하고 해석하였으며 DCT 질감 정보를 이용하여 객체의 영역을 재설정 하였다.

물론 압축된 MPEG2비디오에서 얻을 수 있는 정보들은 최대 블록 단위가므로 객체의 정의도 블록단위 이상의 객체로 제한하였다. 제안된 방법은 비디오 스트림에서 직접 정보를 얻음으로써 계산속도의 향상은 물론, 카메라의 움직임특성과 움직이는 객체의 추적등을 활용하여 기존의 내용기반의 검색 및 분석에도 많이 응용될 수 있다.

이러한 개발 기술들은 압축된 데이터의 검색 및 분석에 유용하게 사용되리라고 기대되며, 특히 검색

틀이나 비디오 편집 툴 또는 교통량 감시 시스템 혹은 무인 감시시스템등에서 압축된 영상의 저장과 빠른 분석을 요구시 필요하리라고 기대된다.

1. 서론

컴퓨터의 발달과 인터넷의 확산으로 멀티미디어의 분야가 급속히 발전하고 있다. 여기에는 원거리 화상회의 및 감시시스템, 주문형 비디오(VOD), 주문형 뉴스(NOD), 디지털 편집 시스템 등 동영상도 포함되어 있다. 하지만 이와 같은 서비스를 제공하기 위해 한 동영상에 가지는 방대한 양의 데이터를 MPEG과 같은 압축형태로 압축하여 저장하거나 전송한다.

본 논문에서는 이러한 압축된 데이터들의 정보를 효율적으로 이용하여, 검색 및 그와 관련된 데이터 베이스의 구축에 활용되는 카메라 움직임의 분석 및 움직이는 객체 분석을 다루고자 한다. 이러한 접근 목적을 위해서 기존의 방법들은 압축된 동영상 데이터를 복호화 한 후 시공간 영역에서 처리하였는데 이러한 비 압축영역에서의 처리는 속도 저하는 물론, 부호화 과정에서 발생하는 유용한 정보 즉 움직임 정보나 블록의 질감 정보 등의 정보를 움직임 추정을 위해 다시 계산하여야 한다는 단점을 내포하고 있다.

한편 이러한 단점을 극복하고자 압축영역에서의 정보를 추출하여 이용하고자 하였는데, S-F Chang 등은 DCT (Discrete Cosine Transform) 영역에서 질감을 처리하는 여러 방법을 시도하였다[1]. 또한 V. Kobla 는 MPEG 의 움직임 벡터를 이용하여 압축영역에서 광 플로우 개념을 확립하였다[2]. 그리고 A. Yoneyama 와 Y. Nakajima 는 움직이는 객체를 정의하고 설명하였다[3][6]. 이러한 논문들의 흐름은 압축 영역에서의 정보이용이라는 측면에서 유용한 방향을 제공하였다.

그러나 움직이는 객체를 추정할 때 움직이는 물체의 움직임 벡터와 물체의 질감만으로 움직이는 물체의 추정은 카메라의 움직임으로 인해 많은 왜곡의

본 연구는 교육부 학술진흥재단 2000-042-E00045에 의해 지원되었습니다.

소지를 안고있다.

본 논문에서는 압축 영역에서의 정보들을 이용하여 카메라의 움직임을 분석 정의하고 움직이는 객체를 추정한다. 또한 카메라의 움직임 분석으로 인해 얻어지는 정보를 활용함으로써 객체인식의 정확성을 높이고자 한다.

2 절에서는 카메라 동작을 인식하기 위해 제안한 알고리즘에 대해서 설명하였고, 3 절에서는 움직이는 물체 인식을 위한 방법에 대해서 설명하였고, 4 절에서는 구현된 시스템 및 연구결과에 대해서 논하였다. 마지막으로 5 절에서는 본 논문의 결론 및 향후계획에 대해서 논하였다.

2. 제안한 카메라 움직임 인식 알고리즘

압축된 MPEG 비디오를 이용하기 위해서 먼저 분석기를 이용하여 각 블록의 움직임 벡터 및 DC 계수 등을 복호화 과정을 수행하지 않고 스트림에서 직접 구한 후, 이들을 이용하여 움직임 벡터의 광 플로우 및 카메라의 움직임을 정의하였다.

2.1 움직임 벡터의 광 플로우

MPEG 은 압축효율을 높이기 위해서 부호화하였으므로 움직임 벡터의 성분들이 순서적이지 못하기 때문에 물체를 추적하기 위해선 순서적으로 바꿀 필요성이 있다. 그래서 움직임 벡터가 있는 각 매크로블록(Macro Block)들을 그림 1 과 같이 프레임의 압축방법에 따른 움직임 벡터의 역할에 따라, 순서적으로 연결한 움직임 벡터의 광 플로우를 구하였다.

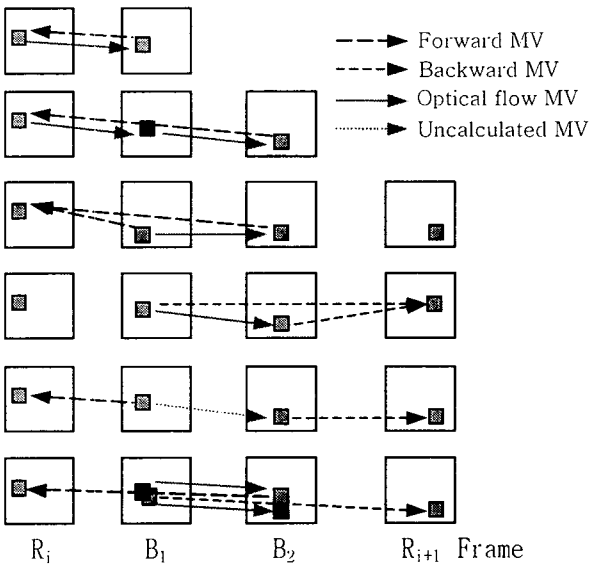


그림 1. 움직임 벡터의 광 플로우

그림 1 에서의 B 프레임처럼 I 와 P 프레임도 같은 방법으로 광 플로우를 구할 수 있다. MPEG 비디오가 원래의 압축방향을 목적으로 움직임 벡터를 예측하고 부호화 하였으므로, 위의 결과는 이상적인 광 플로우는 차이가 있다. 그래서 이러한 단점을 보완 하고

자 구해진 광 플로우를 그림 2 와 같이 3*3의 크기를 가진 윈도우를 가지고 시간적 공간적으로 매디안 필터링을 위한 방법을 도시하였다.

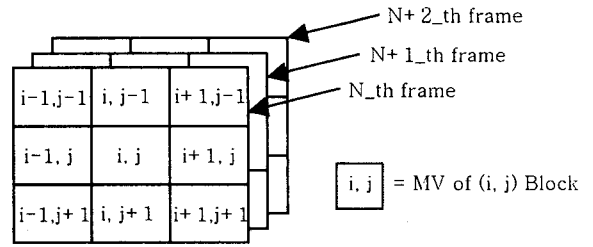


그림 2. 움직임 벡터의 3*3의 매디안 필터링

2.2 카메라 움직임을 위한 알고리즘

카메라의 움직임은 Affine 모델과 Perspective-Projection 모델을 이용하여 근사화 할 수 있다. 그러나 이러한 접근은 압축정보에서 구할 수 없는 카메라의 초점거리를 사용해야 하므로 본 논문에서는 Pan-Tilt Zoom 모델을 사용했다[3][4].

이러한 모델은 R.Milanese 등이 프레임의 대표 값인 Δx , Δy , α 을 구하기 위해서 사용하였는데, 이들은 Leven berg-Marquardt 알고리즘을 반복적으로 사용하여 프레임의 대표 값들을 구하였다. 그러나 본 논문에서는 실행시간이 많이 요구되는 반복적인 계산을 피하고, 불확실한 움직임 벡터의 영향을 줄이고자 Hough 변환을 이용하여 카메라 움직임을 정의하는 매개변수를 구하였다. 또한 이러한 정보는 다음절에서 언급하는 객체인식에 사용하였다.

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. 그리고 α 는 줌의 중심좌표를 중심으로 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 α 로 정의하였다.

그림 3 의 (a)는 카메라가 팬, 틸트, 줌 동작이 발생시 이상적인 매개변수의 분포도이다. 또한 (b)는 실제영상에서 구한 매개변수의 분포도이다.

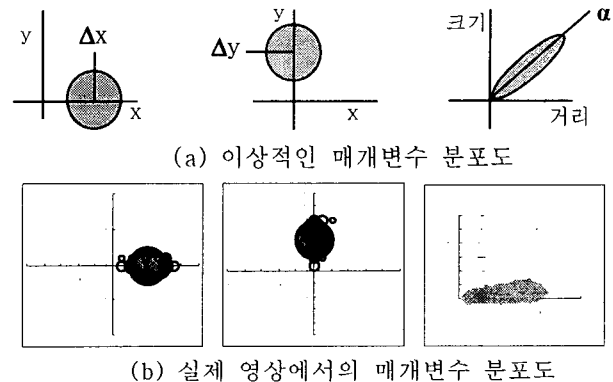


그림 3. 팬 틸트 줌의 매개변수 분포도

즉 카메라 동작을 정의하기위한 Δx , Δy , α 의 매개변수는 시간적인 필터링을 통해 프레임마다 계산되어지며, 연속적인 카메라 움직임을 추정할 수 있다.

또한 α 값을 계산하기위한 중심좌표는 매개변수 값을 결정하는데 중요한 요인이 되는데 측정에러를 줄이고자 각 프레임마다 가로와 세로방향으로 움직임 벡터의 y 와 x 성분이 없는 매크로 블록들에게 가중치를 주어, 누적하여 중심좌표를 구하는 방법을 취하였다. 그림 4 는 확대(Zoom-In)가 발생할 때 각 매크로 블록들의 움직임 벡터의 상태와 가중치를 표현하고, 가장 많이 누적된 중심의 x 좌표를 찾는 그림의 예이다. 중심의 y 좌표도 이와 동일하다.

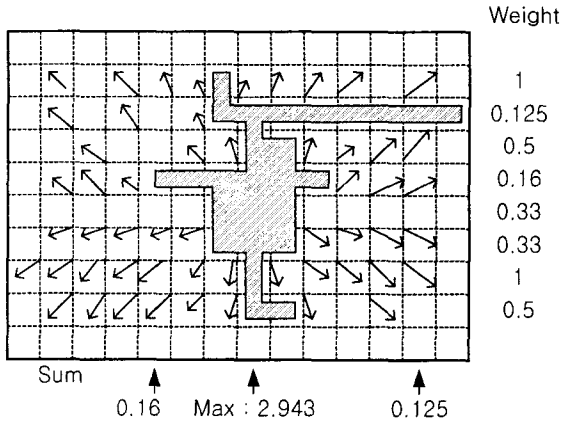


그림 4. 줌의 중심좌표 중 x 성분 구하는 예

3. 제안한 객체 인식 알고리즘 구현

움직이는 물체의 추정은 사용자에게 의해 지정된 물체로부터 물체가 영상에서 사라질 때까지 물체를 추적하는 방식[7]으로 블록의 DC 계수와 움직임 벡터의 광 플로우, 그리고 카메라 동작에 의한 마스크 이미지 등을 조합하여 물체를 추정하는 방식을 택했다.

3.1 카메라 동작영역 마스크 적용

탐색영역의 범위인 f_code 값이 충분한 범위내에서 부호화 된 MPEG 비디오 데이터인 경우, 카메라의 동작속도와 객체 동작속도에 따라 카메라 동작영역의 블록 움직임 벡터와 물체의 블록 움직임 벡터의 기여도가 다를 수 있다. 즉 카메라의 움직임이 너무 빠르면 주위의 배경은 뭉개지고 움직이는 객체만이 질감과 약간의 움직임 벡터 성분을 갖는 경우와 카메라의 움직임이 빠르지 않으면 주위배경 블록의 대부분은 카메라 동작의 움직임 벡터를 갖는 경우로 나눌 수 있다. 전자의 경우는 다음절에서 언급하는 객체의 속성을 이용한 추정방법을 사용하고, 후자의 경우 움직이는 물체는 카메라의 동작영역의 마스크 이미지를 이용하여 배경블록을 제거 후 전자의 방법을 행함으로써 인식의 정확도를 향상시킬 수 있다. 즉 팬과 틸트인 경우 객체를 제외한 배경영역의 블록이 팬과 틸트의 카메라 동작에 해당되는 Δx , Δy 값의 움직임 벡터를 가지게 된다. 그러므로 팬과 틸트에 해당되는 카메라 동작영역을 제거할 수 있다. 또한 줌인 경우

α 값(기울기)에 따라 확장 또는 축소정도를 알 수 있는데 전 프레임의 객체에 이를 적용하여 현재 프레임에서 객체의 대략적인 위치와 크기를 예측하여 줌인 영역을 제거한다. 이러한 전처리 과정을 함으로써 카메라움직임과 독립적으로 객체를 인식할 수 있다.

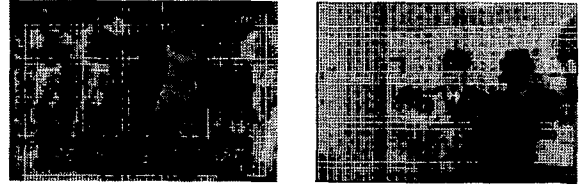
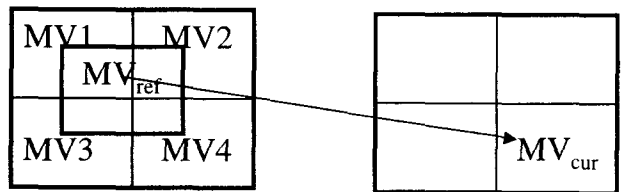


그림 5. 팬과 줌의 마스크 이미지

그림 5는 팬, 틸트와 줌이 발생하는 부분에서 카메라 동작영역을 제거하고 객체가 될 가능성이 있는 블록을 도식한 것이다.

3.2 객체의 속성을 이용한 객체 추적

움직이는 객체의 속성에는 객체의 크기 및 무게 중심, 객체의 움직임 벡터등을 이용할 수 있다. 특히 질감특징 중 DC 성분은 I 프레임에서만 이용 가능하다. 먼저, I 프레임에서만 얻어지는 각 블록의 DC 값들을 취하였다. 그 뒤에 한 GOP 동안 객체 영역으로 설정된 부분에서의 MV 들을 면적 기여도에 따라 누적하였는데 방법은 다음과 같다.



$$MV_{cur} = \sum_{i=1}^4 \frac{h_i w_i}{256} MV_i, \quad MV = \sum_{j=1}^{15} MV_{cur j}$$

그림 6. 한 GOP 동안 누적 움직임 벡터 값을 구하는 예

면적 기여도별로 누적 시킨 움직임벡터 값에서 누적 값이 8pixel 이상의 값이 나오면 주변의 MB 도 객체가 반절 이상 걸쳐져 있는 형태를 가지므로 그 주변의 MB 도 객체의 영역으로 확장을 하였다[7]. 또 12pixel 이상 움직였을 경우는 해당 MB 에서 객체가 3/4 이상 빠져나간 형태이므로 해당 MB 를 객체의 영역에서 제거하였다. 한 GOP 가 지나면 새로운 I Frame 에서 추정된 객체의 영역을 중심으로 후보 영역들을 설정한다. 설정된 후보 영역들과 기존의 I Frame 에서 취했던 DC 값들을 서로 Block Matching Algorithm 을 취해 가장 비슷한 부분을 찾아 새롭게 객체 영역을 설정한다.

4. 제안한 알고리즘의 시스템 구성 및 결과

앞의 2, 3절에서 언급한 알고리즘을 바탕으로 구성된 시스템의 개요도는 그림 7과 같다.

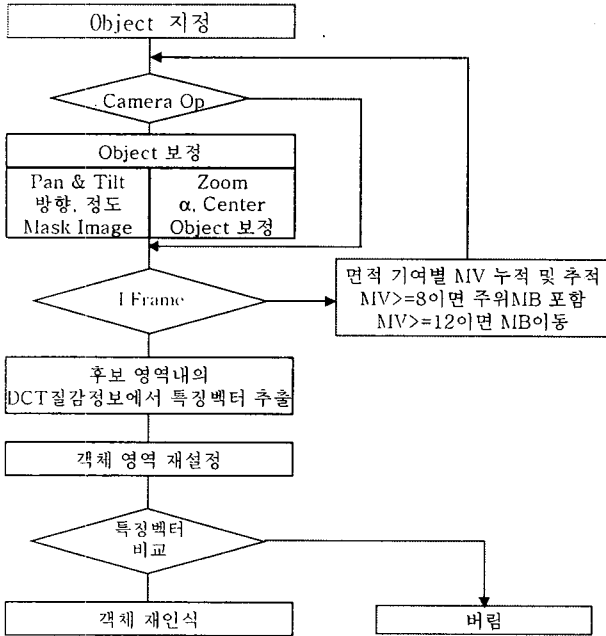


그림 7. 시스템 개요도

또한 구성된 시스템은 결과를 알아보기 위하여 따로 복호화하여 인식결과를 재생할 수 있도록 윈도우 환경의 사용자 인터페이스를 가진 툴로 개발하였다.

다음 그림 8은 46프레임에 대해 객체 추적을 시행한 결과이다.

또한 그림 9는 57프레임에 대해 객체의 움직임 방향을 추적한 결과로 객체가 어떻게 가고 있는지를 보여주고 있다.



그림 8. 움직이는 객체 추적



그림 9. 객체의 움직임 추적 및 해석

5. 결론

본 논문에서는 압축영역에서 객체를 인식할 때 카메라의 동작을 고려하여 객체를 추적함으로써 객체 인식의 정확성을 보완하고자 하였다. 또한 객체의 움직임 벡터를 압축영역에서 바로 얻음으로써 기존의 디코딩된 영역에서 하던 일들에 비해 빠른 속도의 향상을 가져왔다.

이러한 연구는 압축된 비디오의 분석 및 검색에 많이 응용되리라고 기대된다.

향후 객체인식의 정확도를 높이기 위한 후처리 과정 및 알고리즘 개선과 다양한 압축률과 탐색영역을 가진 압축 데이터에도 적용할 수 있도록 알고리즘을 개발하려고 한다.

참고 문헌

- [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] R. Milanese, F. Deguillaume, A. Jacot-Descombes "Efficient Segmentation and Camera Motion Indexing of Compressed Video". *No.rtim.1998.0138*
- [5] Maurizio Pilu, "On using raw MPEG motion vectors to determine global camera motion". *SPIE* vol.3309 p.449-459, 1998
- [6] 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
- [7] Favalli, L., Mecocci, A., Moschetti, F. "Object tracking for retrieval applications in MPEG-2", *Circuits and Systems for Video Technology*, IEEE Transactions on, vol. 10, pp. 427-432, April 2000