

프레임을 변환을 위한 H.264코덱의 움직임 벡터

분석

김상철^o 정현종, 송인선, 남종호

서강대학교 컴퓨터공학과

smaslayer1@nate.com, fona@nate.com, insun0813@naver.com, jhnang@sogang.ac.kr

Analyzing of Motion Vector in H.264 Codec For Frame

Rate Up Conversion

Sangchul Kim^o Hyunjong Jung, Insun Song, Jongho Nang

Dept. of Computer Science and Engineering, Sogang University

요 약

최근 스마트 기기의 보급과 무선 인터넷망의 보급으로 언제 어디서나 비디오를 시청할 수 있다. 하지만 무선 인터넷 망의 품질이 안좋을 경우 영상의 QoS(Quality of Service)를 낮춰 프레임을 스킵하여 전송하게 된다. 이 때 FRUC(Frame Rate Up Conversion)기술을 적용한다면 원본의 프레임 레이트를 확보할 수 있어 QoS를 높일 것으로 기대한다. FRUC에서 MV(Motion Vector)추정시에 연산량이 매우 높아서 스마트 기기에 적용하는 것이 어렵지만 H.264코덱으로 인코딩된 동영상은 자체적으로 MV정보를 갖고 있기 때문에 이 MV를 FRUC에 적용할 수 있다면 FRUC의 연산량을 줄일 수 있을 것이다. 이를 위해서 H.264에 적용된 ME(Motion Estimation)와 FRUC에 적용된 ME의 차이를 고려하여 H.264코덱의 MV가 유용한지 분석하는 것이 선행되어야 한다. 본 논문에서는 H.264 MV와 FRUC의 MV의 차이를 분석하고 유용성을 판단하는 실험을 통해 H.264로 인코딩 된 비디오의 MV중 상당수가 FRUC에 적합함을 확인했다.

1. 서 론

최근 무선 통신망의 확산과 스마트 기기의 보급으로 인해 무선 환경에서 스마트 기기를 통해 비디오 자료를 시청할 수 있다. 하지만 아직까지 제약적인 무선 통신망의 속도로 인해 잦은 버퍼링이나 프레임 스킵을 통해 비디오 자료를 전송받고 시청하게 되는데 이는 사용자 편의성을 크게 떨어트리는 요인 중 하나다. 대형 패널 TV에서는 가상의 프레임을 삽입하여 프레임 레이트를 높여 Motion Blur나 Motion Judder현상을 줄이기 위한 기술로 FRUC(Frame Rate Up Conversion)을 이용한다[1]. 프레임 스킵된 비디오 자료를 전송받아 FRUC를 통해 스킵된 프레임을 복원하여 원본의 프레임 레이트로 재생할 수 있다면 사용자 편의성을 높일 수 있을 것이다. 하지만 FRUC를 위해 사용되는 알고리즘에서 MV(Motion Vector)를 찾기위한 ME(Motion Estimation)과정의 연산량이 높아서 하드웨어에 의존한다[2][3]. 따라서 CPU파워가 낮은 스마트 기기에서 이러한 알고리즘을 그대로 적용하는 것은 불가능하다. MC-DCT기반의 동영상 압축 방법들은 그 자체적으로 MV를 포함한 데이터를 전송 받기 때문에 이 MV 정보를 FRUC에 적용할 수 있다면 연산량을 대폭 줄일 수 있어 스마트 기기에 적용할 수 있을 것으로 예상된다. 이를 위해선 FRUC에 적용 가

능한 MC-DCT기반에서의 MV를 구별하는 방법이 선행되어야 할 것이다. MC-DCT기반의 ME는 기본적으로 연속된 프레임의 차이의 MV를 추정하는 방식을 취하고 FRUC의 ME는 디코딩된 YUV 채널 데이터에 대해 MV를 추정하는 방식이므로 두 방법간의 차이에서 생기는 MV의 특징을 분석하여 이를 실제 FRUC에 적용가능한지에 대해 논의한다. 본 논문의 구성은 다음과 같다. 2장에서는 관련된 연구에 대해 설명하고 3장에서는 FRUC를 위한 H.264의 특징에 대해 설명한다. 4장에서는 실험 분석을 통해 유용성을 확인 하며 5장에서 결론 및 향후 연구로 맺는다.

2. 관련 연구

2.1 프레임 변환 (Frame Rate Up Conversion)

FRUC란 실제 존재하는 프레임들 사이에 가상의 프레임을 생성하여 보여줌으로서 동영상을 보다 자연스럽게 보여주는 효과를 나타내는 기술이다. FRUC는 프레임간의 차이가 나는 부분을 그 부분의 차이를 적용하여 보간 하기 때문에 화면이 심하게 뭉개지는 현상이 적은 MC-FRUC(Motion Compensation Frame Rate Up Conversion)방법이 적용되고 있으며 이 때 MV(Motion Vector)를 추출하는 ME(Motion Estimation) 단계와

ME단계에서 추출된 MV를 활용하여 이미지를 생성하는 MCI(Motion Compensation Interpolation)단계로 나뉜다. ME방법에는 BMA(Block Matching Algorithm), PRA(Pixel Rotation Algorithm)가 있다[4].

PRA는 픽셀 단위로 주변부를 탐색하여 가장 비슷한 픽셀로의 궤적을 움직임 벡터로 구하는 방법으로 정확한 모션 벡터의 추정 가능성이 가능하다. 잡음에 취약할 수 있으며 픽셀별로 검색 영역을 모두 비교해야 하기 때문에 연산량이 높은 단점이 있다.

BMA는 일정 크기의 블록 단위로 주변부를 탐색하여 가장 비슷한 블록 영역으로의 궤적을 움직임 벡터로 구하는 방법으로서 PRA에 비해 정확하지 않은 모션 벡터를 추정하나 블록단위의 비교이기 때문에 잡음에 강하다는 특징이 있다. 또한 블록 단위로 검색 영역을 비교하기 때문에 PRA에 비해 연산량이 낮은 장점이 있다.

대부분의 연구는 연산량의 문제와 잡음에서의 강점 때문에 BMA방법에 대해 연구가 진행되고 있다[5].

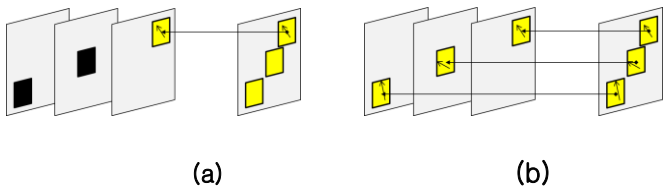


그림1 기존 MPEG의 단일 프레임 참조(a)와 H.264의 복수 프레임 참조(b)

2.2 H.264의 Motion Vector

H.264는 비디오 압축/부호화 표준방식의 하나로 기존의MC-DCT(Motion Compensation Discrete Cosine Inter Prediction Transform)방법을 적용한 비디오 압축/부호화 기술 표준(H.263이나 MPEG4-simple profile)들에 비해 압축 효율 50% 이상 우수하다. 보다 정밀하고 많은 수의 MV를 추정하는 인터 예측(inter prediction) 부호화 방법과 동일프레임에서 공간영역을 예측한 인트라(intra prediction) 예측 부호화 방법 등을 적용하여 인트라 부호화 블록의 영역을 최대한 줄여 압축률을 높였기 때문이다.[6] H.264의 ME는 다음과 같은 특징을 갖는다.

-가변블록 크기의 ME : 16x16~4x4크기의 매크로 블록을 이용하여 ME를 수행한다.

-복수 참조 픽처로부터의 예측 : <그림 1>처럼 MPEG-2, MPEG-4의 P-frame에서는 바로 이전에 위치한 I-frame이나 P-frame을 참조픽처로 사용하지만 H.264는 복수 참조 픽처가 가능하여 이를 통해 최적의 블록을 선택할 수 있어서 보다 많은 인터 예측이 가능해 높은 압축 효율을 갖는다.

3. FRUC를 위한 H.264의 특성

인트라 블록 vs 예측 블록

인트라 블록의 경우 참조 프레임들과의 차이가 너무 커서 MV

로 나타내기에 부적합할 경우 해당 프레임에서 블록안의 이미지를 그대로 보내는 블록이다. 인트라 블록은 MV정보가 없기 때문에 디코딩 된 후에 MV를 추출하기 위해 기존의 ME를 적용해야 한다. 또한 예측 블록은 MV정보를 포함한 인터 예측 블록과 주변부 블록과 유사관계가 있다는 정보만 포함한 인트라 예측 블록으로 나뉜다. H.264의 ME와 FRUC의 ME 방법에 근본적인 차이로 인해서 인트라 예측 블록에 저장된 MV가 FRUC를 적용할 때 얼마나 활용이 가능한지 확인해 보아야 한다.

3.1 인트라 예측

인트라 예측은 동일 프레임 내에서 주변 픽셀부위의 색상은 비슷하다는 특징을 이용하여 JPEG 압축방식에 사용되는 Spatial Coding기법을 이용하여 주변부 블록을 이용하여 자신 블록의 색상값을 예측하는 방식으로 이러한 블록을 인트라 예측 블록이라 한다. 인트라 예측 블록은 자체적으로 모션벡터를 가지고 있지 않지만 자기 주변부 블록의 값을 그대로 가져올 수 있으므로 8번의 비교만으로 MV를 적용할 수 있다. 하지만 인트라 예측 블록이 위치할 곳을 정하기 위해서는 주변부에 위치한 예측블록에서 나오는 MV에 의존적이다.

3.2 인터 예측

인터 예측은 실제 연속된 프레임간에 현재 위치가 다음 프레임에서 어느 방향으로 움직였는지를 예측하는 방법이다. P-frame은 현재 프레임으로부터 이전의 여러 I-frame 과 P-frame을 참조하고 B-frame은 자신과 인접한 I-frame과 P-frame만을 참조한다. FRUC에서는 참조 프레임간의 정보를 이용하여 현재 블록을 기준으로 주변부에 가장 유사한 참조 프레임에서의 블록 위치를 찾는 반면 H.264에서는 주변부에 가장 유사한 차이가 나는 부분 중 비트레이트가 가장 낮은 블록 위치를 찾는다. 이러한 차이점 때문에 H.264의 MV를 그대로 가져다 쓸 수 없다. 따라서 유용한 MV는 H.264 비디오의 MV와 실제 Full Search를 통한 MV간의 차이가 적은 것이라 할 수 있기 때문에 유용한 벡터를 식(2)와 같이 정의했다.

$$Diff(B_i) = |H_B MV_x - R_B MV_x| + |H_B MV_y - R_B MV_y| \quad (1)$$

식(1)은 H.264 Decoder에서 추출된 프레임의 i번째 블록의 $H_B MV$ 와 Raw Data에서 ME를 통해 얻어진 $R_B MV$ 간의 차이이다. 차이가 4 이하의 H.264 Decoder의 MV는 정확하지 않은 MV라 하더라도 나중에 4번의 블록간 차이의 연산을 구하면 되므로 ME에서 수행되는 Full Search의 비교 연산량인 Search Range가 N일 때 $O(N^2)$ 에 비해 매우 적다고 할 수 있다.

$$\begin{cases} Diff(B_i) \leq 4, H_B MV \rightarrow Useful \\ Diff(B_i) > 4, H_B MV \rightarrow NoUseful \end{cases} \quad (2)$$

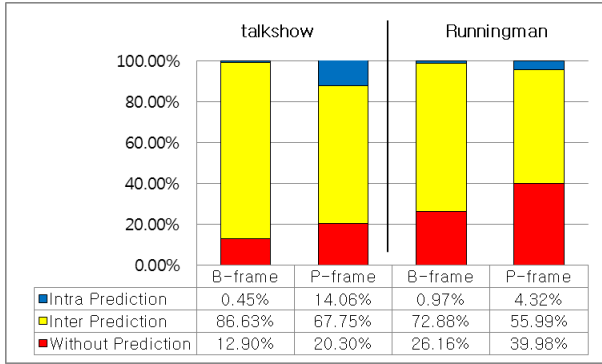
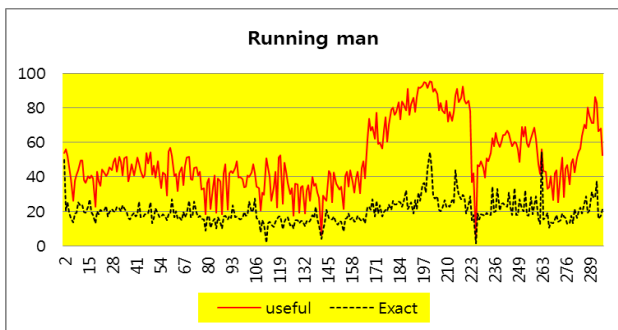


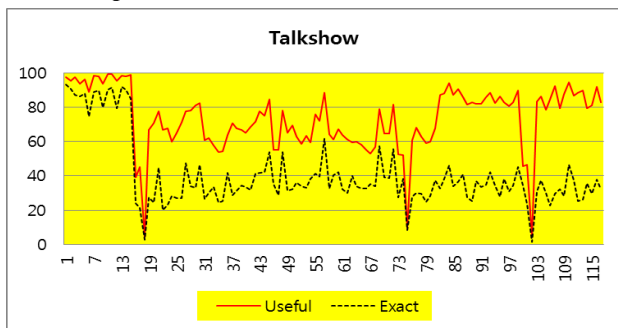
그림2 runningman 영상과 talkshow영상의 예측 블록 비율

4. 실험 결과

H.264로 인코딩 된 비디오의 MV의 유용성을 알아보기 위해 다음과 같은 환경으로 실험을 하였다. 움직임이 많은 영상(runningman)과 움직임이 적은 영상(talkshow)에서의 H.264 MV와 디코딩된 프레임 이미지를 이용하여 H.264의 인터 예측 블록에서 실제 ME를 수행하였다. 그림2는 움직임이 많은 runningman영상과 talkshow영상의 모든 프레임에서 나타난 예측 블록과 비예측 블록의 비율을 나타낸다. 일반적으로 움직임이 적은 talkshow영상에서 예측 블록의 부분이 많이 나타난다. 이는 MV를 활용할 수 있는 부분이 많은 것으로 FRUC를 적용할 때 시간 감소를 시킬 수 있는 부분으로 고려할 수 있다. 일반적으로 미래의 프레임까지 고려하는 B-Frame의 경우 인터 예측 블록의 비율이 높다. 움직임이 많이 나타나는 영상에서는 프레임간 차이가 크기 때문에 영상의 차이를 그대로 보내는 비예측 영역이 높게 나타나는 것을 알 수 있다.



(a) runningman영상의 유용한 MV와 일치하는 MV의 비율



(b) talkshow영상의 유용한 MV와 일치하는 MV의 비율

그림3

그림3은 각 프레임에서 추출된 인터 예측 블록의 MV중 유용한 MV와 실제 프레임에서 ME로 구한 MV들과 일치하는 MV의 비율을 나타낸 그래프이다. 움직임이 많은 영상에서 유용한 MV는 52.4%의 비율을 나타내고 움직임이 적은 영상에서 유용한 MV는 72.8%의 비율을 나타낸다. 실제 스포츠 영상을 제외한 대부분의 영상은 움직임이 크게 변화하는 장면이 많지 않으므로 H.264 MV를 활용하여 FRUC에 적용한다면 유용할 것이다.

5. 결론 및 향후 연구 방향

실험을 통해서 움직임이 많지 않은 영상에서 상당수의 H.264의 MV가 FRUC에 유용함을 밝혔다. 또한 상당수의 영상은 움직임이 크게 변화하는 장면이 많지 않기 때문에 위의 MV를 활용하여 저연산량의 FRUC방법을 고안할 수 있을 것이다. 대체적으로 주변부의 블록은 유사한 방향과 크기의 MV를 갖기 때문에 비예측 부분과 유용하지 않은 부분에서 H.264 영상의 MV를 활용하여 저연산량의 MV를 추정하는 방법과 이렇게 추정된 MV를 통해서 기존 FRUC의 문제점인 홀 현상이나 블록킹 현상 없이 가상의 프레임을 생성하는 방법에 대해 연구할 것이다. 또한 유용한 MV를 미리 판별할 수 있는 방법에 대한 연구도 필요하다.

6. 참고 문헌

[1] Suk-ju Kang, Kyoung-Rok Cho, Young Hwan Kim, "Motion Compensated Frame Rate Up-Conversion Using Extended Bilateral Motion Estiamtion," *IEEE Transaction on Consumer Electronics*, vol. 53, no.4, pp.1759-1767, 2007.

[2] Demin Wang, Liang Zhang, Vincent, A., "Motion-Compensated Frame Rate Up-Conversion-Part I: Fast Multi-Frame Motion Estimation," *IEEE Transaction on Broadcasting*, vol. 56, no.2, pp.133-141, 2010.

[3] Demin Wang, Vincent, A., Blanchfield, P., Klepko, R., "Motion-Compensated Frame Rate Up-Conversion-Part II: New Algorithms for Frame Interpolation," *IEEE Transaction on Broadcasting*, vol. 56, no.2, pp.142-149, 2010.

[4] H.S. Kang, "The Research for the Caption Processing In The Frame Rate Up-Conversion," *Master Thesis, Hanyang University*, Seoul, Korea, 2011. (in Korean)

[5] Bo-Won Jeon, Gul-Il Lee, Sung-Hee Lee., Rae-Hong Park., "Coarse-to-Fine-Frame Interpolation for Frame Rate Up Conversion Using Pyramid Structure," *IEEE Transaction on Consumer Electronics*, Vol49, No.3, pp. 499-508, 2003.

[6] Francisco Arévalo López, "Frame Rate Up-Conversion with Frame Interpolation over non-regular block partitions of H.264," *Master Thesis, Royal Institute of Technology, Stockholm, Sweden*, 2008.