

효율적인 동영상 프레임율 상향 변환을 위한 MC-DCT 기반 인코더에서의 움직임 벡터 추정 방법

모홍철^o 김상철, 낭종호

서강대학교 컴퓨터공학과

angel88317@nate.com, smaslayer1@nate.com, jhnang@sogang.ac.kr

A Motion Vector Estimation Method for Efficient Video Frame Rate Up Conversion in MC-DCT based Video Encoder

Hongchul Mo^o Sangchul Kim, Jongho Nang

Dept. of Computer Science and Engineering, Sogang University

요 약

생략된 비디오 프레임을 보간하고자, 디코더 단에서 블록 단위의 움직임 벡터 추정이 수행되는 기존의 MC-FRUC (Motion Compensated Frame Rate Up Conversion) 방법은 디코딩 수행 시 많은 연산을 필요로 하기 때문에 무선 단말기와 같이 자원에 제약된 환경에서는 사용할 수 없다는 문제점을 가지고 있다. 본 논문에서는 MC-DCT 기반 코덱에서의 효율적인 MC-FRUC를 위해 인코딩 수행 시, 디코더 단에서 보간해야 할 프레임을 미리 고려한 움직임 벡터 추정 기법을 제안한다. 제안한 방법에서는 인코더 단에서의 프레임 간 코딩 (Inter-Frame Coding) 시 원본 프레임 정보를 이용하여 원본 프레임과의 차이를 가장 작도록 만드는 보간 프레임을 생성하는 움직임 벡터를 그리디 (Greedy) 방식으로 탐색한다. 이렇게 생성된 움직임 벡터는 디코더 단에서의 MC-FRUC 수행 시 바로 사용가능하기 때문에, 추가적인 연산 없이도 생략된 원본 프레임과 가장 유사한 보간 프레임을 생성할 수 있도록 해준다. 다양한 종류의 동영상에 대한 실험을 통하여, 제안한 방법에 의해 인코딩된 동영상은 디코더 단에서의 추가적인 연산 없이도 높은 화질의 보간 프레임을 MC-FRUC 방식으로 생성할 수 있음을 알 수 있었다.

1. 서론

스마트 기기의 보급 증가와 향상된 무선 인터넷 환경에 따라 누구나 스마트 기기를 통해 언제 어디서든지 웹에 접근할 수 있게 되었고, N-Screen 서비스, 영상 통화 등 다양한 영상 관련 서비스들을 이용할 수 있게 되었다. 하지만 여전히 외부에서의 불안정한 네트워크 환경과 속도의 제약, 멀티미디어 데이터가 갖는 대용량의 특성 등으로 인해 낮은 대역폭을 가진 네트워크 환경에서는 데이터 전송을 위해 영상의 프레임율을 낮추어 전송을 하게 된다. 하지만 낮은 프레임율의 동영상의 경우, 영상을 받는 수신 측에서는 동영상이 끊기며 재생되는 듯한 느낌을 받게 되며, LCD, LED TV의 경우 낮은 주사율로 인해 모션 저더(Motion Judder), 모션 블러(Motion Blur) 현상이 발생하여 시청자의 눈의 피로도를 높이는 등의 문제가 발생하게 된다[1]. 이러한 문제들을 해결하기 위해 디코더 단에서 가상의 프레임(Frame)을 삽입하는 기술인 FRUC (Frame Rate Up-Conversion)를 적용하게 된다[2]. 이런 기술 중에서 MC-FRUC(Motion Compensation-Frame Rate Up Conversion) 기술은 적은 연산량과 잡음에 강건하여 가장 널리 사용되고 있는 기술이나[2], 무선 단말기 등에서는 여전히 적용하기 힘든 연산량을 갖고 있는 문제점이 있다. 본 논문에서는 MC-DCT(Motion Compensation-Discrete Cosine Transform) 기반의 동영상 코덱에서 디코더단에서 FRUC를 빠르게 수행할

수 있도록 하는 인코더단에서의 움직임 벡터 추정 방법을 제안한다. 제안한 기본 방법에서는 움직임 벡터에 의해 연관된 인접한 프레임들의 두 블록을 가지고 보간을 수행한 보간 블록과 원본 프레임 내 같은 위치에 있는 블록과의 차이를 가장 작게 만들어주는 움직임 벡터를 그리디 방식으로 탐색한다. 하지만 이런 기본 방법은 크게 두 가지 문제를 가지고 있는데, 현재 프레임 내 블록이 여러 개의 움직임 벡터를 가지게 될 수 있다는 것과 보간 프레임 생성에 이용되지 않는 블록이 생기게 되어 움직임 벡터가 없는 블록이 생성될 수 있다는 점이다. 이런 문제를 해결하기 위하여 본 논문에서는 추정된 움직임 벡터가 만족하여야 할 제한 조건을 제안하였으며, 이런 제한 조건을 만족하는 움직임 벡터의 집합을 최종 인코딩 결과로 출력하도록 하였다. 본 논문에서 제안한 움직임 벡터 추정 방법은 처음부터 인코더 단에서 FRUC에 최적화된 움직임 벡터를 생성하기 때문에 디코더단에서 추가적인 계산 없이 FRUC를 할 수 있는 장점을 가지고 있으며 실험을 통하여 기존의 FRUC보다 좋은 화질을 얻음을 보였다. 본 논문의 나머지 장들은 다음과 같이 구성된다. 2장에서는 FRUC에 최적화된 움직임 벡터 생성을 위한 MC-DCT 기반 코덱에서의 움직임 벡터 추정 기법에 대해 설명한다. 3장에서는 제안한 방법에 대한 실험을 통해 성능 및 특성을 비교 분석하였다. 4장에서는 결론 및 향후 연구 방향에 대해 기술한다.

2. 관련 연구

2.1 프레임 상향 변환

FRUC 란 실제 존재하는 프레임들 사이에 가상의 프레임을 생성하여 보여줌으로서 동영상을 보다 자연스럽게 보여주는 효과를 나타내는 기술이다. 이 중 연산량과 화질을 모두 고려한 MC-FRUC(Motion Compensation Frame Rate Up Conversion)방법이 적용되고 있다. MC-FRUC 는 두 프레임의 움직임을 계산하여 그 중간에 생성될 프레임의 움직임을 예측하여 이를 토대로 프레임을 생성하는 기법이다.

2.1.1 단방향 프레임 상향 변환

현재 프레임과 참조 프레임의 객체의 움직임을 추정해 움직임 벡터를 추정할 때 참조 프레임이 이전이나 이후 프레임중 한쪽 방향만 참조할 경우에 단방향 프레임 상향 변환(Unidirection FRUC)라 한다[3]. 단방향 프레임 상향 변환은 잘못 추정된 움직임 벡터로 인해 특정 부분에 픽셀값이 입력이 안되는 홀 현상이 발생하는 문제점이 있다[4].

2.1.2 양방향 프레임 상향 변환

생성될 프레임을 기준으로 시간적 대칭성을 고려하여 현재 프레임과 참조 프레임의 움직임 벡터를 추정하여 프레임 양방향을 모두 참조할 경우에 양방향 프레임 상향 변환(Bilateral FRUC)라 한다[3]. 단방향 프레임 상향 변환의 한계점인 홀 현상을 근본적으로 방지하지만 자연스러운 연결됨을 보이기 위해 후처리를 하는 과정에서 영상의 이미지가 흐릿해지는 모션 블러 현상이 나타나는 문제점이 있다[4].

3. 탐욕방식의 FRUC를 위한 MC-DCT기반 움직임 벡터 추정

FRUC를 하기 위해서는 그림 1과 같이 생성될 보간 프레임 f_i 이 생략된 원본 프레임(Original Frame) f_0 과 차이가 가장 적게 하는 움직임 벡터가 필요하다. 따라서 생략될 원본 프레임을 기준으로 현재 프레임 f_n 과 참조 프레임 f_{n-1} 을 타깃으로 brute-force방식으로 움직임 벡터를 찾아야 한다. 하지만 이는 exhaustive search가 되기 때문에 인코딩 연산에 적합하지 않다. 따라서 본 논문에서는 탐욕 방식으로 움직임 벡터를 찾는 방법을 제안한다.

3.1 탐욕방식의 목적함수 정의

3절에서 설명했듯이 연산량을 고려하여 Exhaustive Search를 피하기 위해서 탐욕방식을 적용하는 데 이때에 목적함수 설정이 필요하다. 식1은 움직임 벡터 추정 한 후 해당 움직임 벡터로 FRUC를 수행해 생성하는 블록의 보간 이미지를 나타내는 식으로 $Block(x, y, f_n)$ 은 주어진 프레임 f_n 내의 위치 (x, y) 에 있는 블록을 의미한다. 식2는 제안하는 탐욕방식의 목적함수로서 식1의 이미지와 원본 프레임 영상의 블록을 비교하여 차이가 가장 적게 하는 움직임 벡터를 인코더의 움직임 벡터로 지정하는 것을 나타내는 것으로 (dx, dy) 는 후보 움직임 벡터를 나타내며 $S_{x,y}$ 는 움직임 벡터 후보가 갖는 탐색 범위를 의미한다.

$$MCI' = MCI(Block(x - dx, y - dy, f_{n-1}), Block(x + dx, y + dy, f_n)) \quad (1)$$

$$MV_{x,y}^f(f_n) = argmin_{(dx, dy) \in S_{x,y}} SAD(MCI', Block(x, y, f_0)) \quad (2)$$

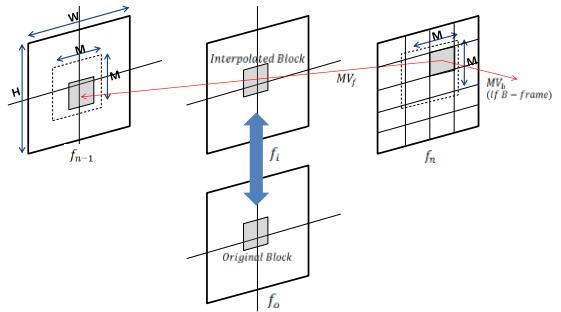


그림 1. FRUC를 위한 MC-DCT 움직임 벡터 추정의 문제 정의

3.2 제약조건 설정

3.1절에서 설명한 목적함수에 의해서 움직임 벡터를 생성하면 참조프레임에서 하나의 블록이 여러 개의 움직임 벡터를 갖거나 움직임 벡터가 존재하지 않는 문제가 발생한다. 따라서 이러한 문제를 해결하기 위해서 제약조건을 설정 해야 한다. 가상 보간 프레임(Interpolated Frame) f_i 내의 블록을 지나는 움직임 벡터가 반드시 하나는 있어야 움직임 벡터를 통해 연관 된 인접한 프레임들의 두 블록을 가지고 보간을 수행할 수 있고, 또한 가상 보간 프레임 블록을 지나는 움직임 벡터가 두 개 이상일 경우에는 겹침 현상이 발생하므로 보간에 대한 왜곡 현상이 발생하게 되는 문제가 있으므로 식 3과 같이 가상 보간 프레임 내 블록을 지나는 움직임 벡터가 반드시 한 개 존재 해야 하는 제약 조건 C_1 을 지정하였다. $n(MV_{x,y}^{through}(f_i)|Block(x, y, f_i))$ 는 f_i 내의 블록 $Block(x, y, f_i)$ 을 통과하는 움직임 벡터 $MV_{x,y}^{through}(f_i)$ 의 개수를 나타낸다.

$$if (n(MV_{x,y}^{through}(f_i)|Block(x, y, f_i)) = 1) then true, otherwise false \quad (3)$$

또한 현재 프레임의 움직임 벡터가 참조 프레임의 여러곳에서 참조될 수 있는 문제점이 있다. 이를 해결 하기 위해서 제약 조건 C_2 를 식 4와 같이 지정하였다.

$$if (n(MV_{x,y}^f(f_n)|Block(x, y, f_n)) = 1) then true, otherwise false \quad (4)$$

여기서 $n(MV_{x,y}^f(f_n)|Block(x, y, f_n))$ 은 현재 프레임 f_n 내의 블록이 포함하는 움직임 벡터 $MV_{x,y}^f(f_n)$ 의 개수를 나타낸다.

4. 실험 및 결과

본 장에서는 3장에서 제시한 알고리즘의 성능을 평가하기 위한 여러 가지 실험 및 분석을 수행하고 이를 통해 본 논문의 알고리즘의 타당성을 검증하였다.

4.1 실험 데이터 및 평가 방법

본 논문에서는 다양한 움직임을 갖는 영상에 대해 프레임 보간을 수행하여 실시간성 가능성과 영상의 품질을 실험 하였다. 각 영상의 모션 에너지(Motion Energy)는 표 1과 같다. 모션 에너지는 모든 프레임에 대해 각 프레임 간 차이들을 계산하여 이에 대한 평균으로 결정하였다[5].

표 1. 각 실험용 영상 별 모션 에너지(Motion Energy) 수치

Video Files	Motion Energy	Video Files	Motion Energy
Run	11.081	Bridge	1.74316
3D Game	8.159	Videophone	1.025
Foreman	5.23617	Desktop	0.112

4.2 프레임 보간 수행 시간 분석

그림 2를 보면 기존의 FRUC(Uni-Direction MV, Bilateral MV) 기법의 경우 기존 디코더 단에서 모든 블록에 대한 움직임 벡터 추정이 수행되어 수행시간이 다른 기법에 비해 오래 걸리는 결과가 나타난다. 그리고 MC-DCT의 움직임 벡터를 그대로 활용하여 FRUC(MC-DCT MV)를 하는 경우와 본 논문에서 제안한 움직임 벡터 추정이 생략되기 때문에 프레임 보간 수행시간이 생략됨을 알 수 있다. 또한 FRUC(DUMP MV)[5]의 경우 FRUC에 유용한 움직임 벡터를 검출하여 프레임 보간시 움직임 벡터 탐색을 부분적으로 생략하기 때문에 수행시간이 기존의 FRUC들에 비해 작지만 비 유용한 움직임 벡터 부분에서의 수행시간 때문에 FRUC(MC-DCT MV)에 비해 연산이 더 걸림을 알 수 있다.

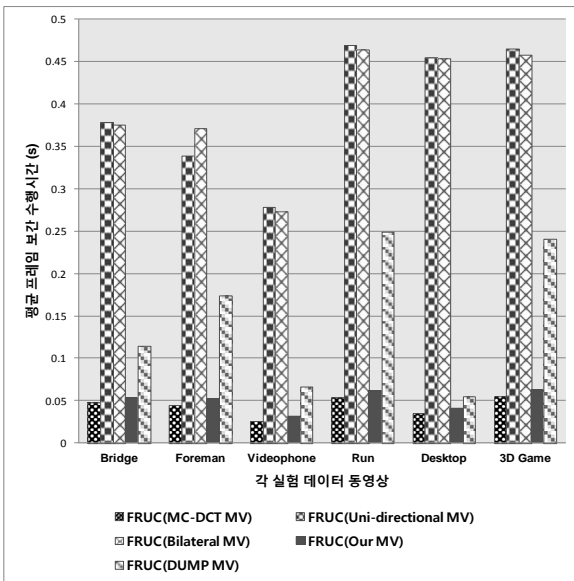


그림 2. 프레임 보간 수행 시간 비교 표

4.3 프레임 보간 화질 분석

그림 3은 프레임 보간 화질을 비교하기 위하여 기존의 기법들에 대해 PSNR값을 측정하여 표이다. 비트레이트 최적화를 위해 생성된 움직임 벡터를 활용하여 프레임을 생성할 경우 FRUC에 적합하지 않기 때문에, PSNR이 가장 떨어짐을 알 수 있다. 또한, 기존의 FRUC 기법들에 비해 본 논문에서 제안한 방법의 화질이 높음을 알 수 있다. 이는 기존 FRUC에서 존재하지 않는 생략된 원본 프레임(Original Frame) f_0 정보를 활용하여 FRUC에 적합한 움직임 벡터를 생성하였기 때문이다. Video Phone이나 Desktop 환경에서는 움직임이 적기 때문에 Bilateral-FRUC가 유리하기 때문에 제안한 방법이 PSNR이 적게 나왔다. 하지만 PSNR이 40에 근접하기 때문에 시청에 불편함이 없다. 기존 DUMP 기법의 경우 화질이 기존의 FRUC에 비해 떨어지는데, 이는 임계치에

의존적인 부분들이 있어 유용한 움직임 벡터를 100% 분류하지 못하기 때문에 특히 움직임이 많은 영상에 대해 약점을 가지고 있기 때문이다[5].

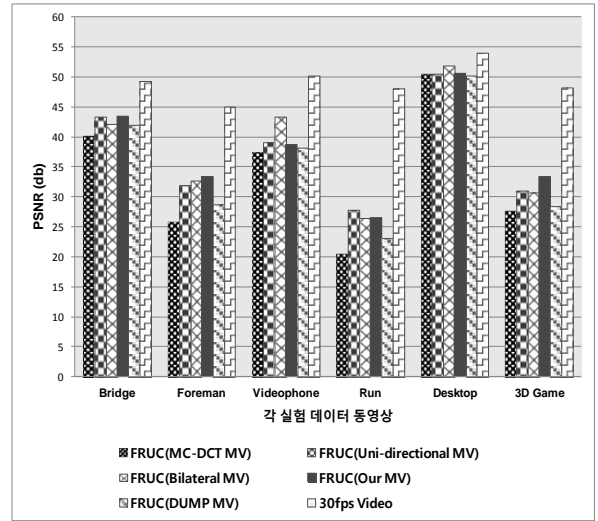


그림 3. 프레임 보간 화질 비교 표

5. 결과 및 향후 연구

본 논문에서 제안한 방법은 인코더에서 FRUC에 적합한 움직임 벡터를 생성하여 단말기 단에서 실시간 FRUC가 가능함을 보였다. 하지만 탐색방식과 인코더의 한계로 인하여 모든 블록에서 최적의 FRUC를 위한 움직임 벡터가 생성되지 못하기 때문에 프레임 보간 작업을 할 때, 제약조건에 걸려진 매크로 블록에 대해 움직임 벡터를 재 추정하는 알고리즘이 필요하기 때문에 이에 대한 연구를 향후에 진행할 것이다.

6. 참고 문헌

[1] B. D. Choi, J. W. Han, C. S. Kim, and S. J. Ko, "Motion Compensated Frame Interpolation Using Bilateral Motion Estimation and Adaptive Overlapped Block Motion Compensation," *IEEE Transactions on Circuits and System for Video Technology*, Vol. 17, No. 4, pp. 407-416, 2007.

[2] S.H. Lee, Y.C. Shin, S. Yang, H.H. Moon, and R.H. Park, "Adaptive Motion-Compensated Interpolation for Frame Rate Up-Conversion," *IEEE Transactions on Consumer Electronics*, Vol. 48, No. 3, pp. 444-450, 2002.

[3] D. W. Vincent, A. Blanchfield, P. Klepko, and R. Commun, "Motion Compensated Frame Rate Up-Conversion Part I: Fast Multi-Frame Motion Estimation," *IEEE Transactions on Broadcasting*, Vol. 56, No. 2, pp. 133-141, 2010.

[4] 김상철, 정현중, 송인선, 낭종호, "프레임을 변환을 위한 H.264코덱의 움직임 벡터 분석," *2012 한국 컴퓨터 종합학술대회 논문집*, 제 39권, 제 1호, 164-166쪽, 2012.

[5] S. C. Kim, "An Useful Motion Vector Classifying Method for Efficient Frame Rate Up Conversion in H.264 Video Stream," Master Thesis, Sogang University, 2011. (in Korean)