

DCT 공간에서 크기를 변환하는 방법을 사용하는 고속 매체변환기

한재웅^o, 채병조, 오승준, 정광수
광운대학교 전자공학부

{jwhan, bongbong, sjoh}@media.gwu.ac.kr and kchung@daisy.gwu.ac.kr

A Fast Video Transcoder Using DCT-Domain Resizing Algorithm

Jae-Woong Han^o, Byung-Jo Chae, Seoung-Jun Oh, Kwang-Su Jung
School of Electronics, Kwangwoon Univ.

본 논문에서는 대역폭이 보장되지 않는 네트워크 상에서 비디오 서비스를 제공하기 위한 매체변환 방법 중 비디오 프레임 크기를 변환시켜 비트율을 효율적으로 제어하는 방법을 제안한다. H.263으로 부호화된 스트림을 DCT 공간에서 크기를 변환시킴으로써 완전 복호화 후 다시 부호화하는 과정에서 생기는 계산량을 대폭 감소시켜 보다 효율적인 방법으로 비트율을 제어할 수 있다. 움직임 정보뿐만 아니라 이전 프레임과의 차이값을 다시 사용하기 위해 크기변환 후 H.263 표준의 향상된 예측 모드(Advanced Prediction Mode)를 사용하여 화질저하도 최소화한다. 실험결과 화질면에서는 블록화 현상(Block Artifact)이 크게 감소하였고, 프레임이 진행됨에 따라 PSNR은 낮아졌지만, 매체변환은 기존의 방법에 비해 빠른 속도로 이루어 질 수 있었다.

1. 서론

향상된 네트워크 기술로 인해 인터넷상에서의 멀티미디어 스트리밍(Streaming) 서비스가 활성화되고 있다[1]. 그럼에도 불구하고 인터넷의 특성상 보장되지 않는 대역폭으로 인하여 원활한 서비스가 제공되지 못하는 경우가 많다. 이러한 환경에서의 실시간 비디오 부호화 응용들에는 비트율 제어(Bit Rate Control)가 서버(Server)측의 비디오 부호기 자체에서 가능하지만, 일반적으로는 서버측에서 부호화된 스트림 자체를 파일 형식으로 가지고 있다. 서버가 실시간 부호화 시스템이라든가 일정한 비트율을 가지는 비디오 스트림만을 제공하는 것은 여러 클라이언트(Client)들의 입장에서 볼 때, 제공되는 대역폭이나 계산능력 등이 모두 다르기 때문에 서비스에 한계를 가져온다.

과거에는 이러한 현상을 방지하기 위하여 여러 계층(Layer)으로 분리된 부호화 방식이 사용되었다. 이 부호화 방식은 비디오 영상을 중요도에 따라 여러 계층으로 분리시킨 후 대역폭에 따라 계층의 일부분이나 전체를 받아보도록 하는 방법이다. 최근에는 이와 달리 멀티미디어 서비스를 제공하는 서버와 서비스를 제공받는 클라이언트 사이에 매체변환기(Transcoder)를 두어 대역폭의 변동에 능동적으로 비트율을 제어할 수 있도록 하는 방법이 많이 사용된다. 매체변환기는 서버가 가지고 있는 멀티미디어 데이터를 원하는 비트율에 부합시키기 위하여 여러 방법을 사용하고 있는데, 대표적으로 양자화 폭을 조절하는 방법[2]과 프레임율(Frame Rate)을 조절하는 방법[3], 그리고 비디오 영상의 크기를 변환하는 방법 등이 있다.

본 논문에서는 비디오 크기를 감소시켜 비교적 낮은 비트율이 요구될 때 이것을 제어하는 방법을 제안하고자 한다. 기존에는 입력 스트림에 대하여 완전 복호화한 다음 영상의 크기를 변환하여 다시 부호화하는 기본적인 방법으로부터, 입력 스트림 정보에서 움직임 벡터(Motion Vector)를 추출하여 다시 재사용함으로써 움직임 검출(Motion Estimation)에 대한 부담을 줄이는 방법까지 여러 알고리즘이 제안되었으나[4], 부분적으로 완전 복호화와 부호화를 한다는 점은 변함이 없었다.

제안된 방법은 입력 스트림을 완전 복호화하지 않고, 가변 길이 복호화(Variable Length Decoding)와 역양자화(Dequantization) 과정 후에 생성되는 DCT(Discrete Cosine Transform) 계수와 움직임 정보로 크기감소된 스트림을 만들어냄으로써, 움직임 검출뿐 아니라 역DCT(Inverse DCT), DCT 과정을 생략하게 하여 매체변환기의 부담을 더욱 줄여서 클라이

언트가 보다 빠른 서비스를 제공할게 하는데 목적이 있다.

본 논문은 다음의 순서로 구성되어 있다. 먼저 2장에서는 DCT 공간에서의 비디오 크기감소 알고리즘을 소개하고, 3장에서는 이 알고리즘을 실제 H.263 스트림에 적용시키는 방법을 제시한다. 4장에서는 여러 종류의 부호화된 비디오영상에 대한 실험결과를 살펴보고, 마지막으로 5장에서 결론을 맺도록 한다.

2. DCT 공간의 비디오 크기감소

한 프레임의 크기를 2:1로 줄이는 방법은 화소 공간(Pixel Domain)에서는 매우 간단한 알고리즘으로 구현이 가능하다. 크기감소는 DCT 공간에서도 가능하며, 특별한 응용에 있어서는 화소 공간의 크기감소 알고리즘 보다 더 효율적이다[5].

보편화된 표준에서 사용하는 블록의 크기는 8×8이고, 휘도 성분만을 고려했을 경우, 블록 네 개가 하나의 매크로블록(Macroblock)을 구성한다. 크기감소는 매크로블록 단위로 이루어진다. 크기감소를 원하는 DCT 공간의 매크로블록 행렬을 M 이라 하고, 감소된 후의 DCT 공간에서의 8×8 블록 행렬을 B 라 하면 두 행렬은 다음과 같은 관계를 갖는다.

$$B = TMT^t \quad (1)$$

식 (1)에서 행렬 T 는 DCT 공간에서 16×16 행렬을 8×8 행렬로 변환시키는 행렬이다. T 의 크기는 8×16이다. M 행렬에 대해 원하는 동작을 하나씩 추가시킴으로써 T 변환행렬을 정의한다.

$$B_F = FMF^t \quad (2)$$

F 는 크기감소 후의 앨리어싱(Aliasing)을 방지하기 위한 저대역 필터(Low Pass Filter) 행렬이다. 즉, 이 행렬의 크기는 16×16이 되고, 다음과 같이 정의된다.

$$F = I_2 \otimes \begin{bmatrix} I_4 & 0 \\ 0 & 0 \end{bmatrix} \quad (3)$$

I_N 은 $N \times N$ 단위행렬이고 \otimes 는 크로네커 곱(Kronecker Product)을 나타낸다. F 는 네 개의 블록 각각의 좌상단의 계수만 남겨두고 나머지 고주파성분을 모두 0으로 만들어서 저대역 필터 역할을 수행하여, 블록화 현상(Block Artifact)을 감소시키는 역할을 함과 동시에 앨리어싱을 방지하여 크기감소 후 화질이 왜곡되는 것을 방지한다.

다음 단계로 저대역필터가 수행된 행렬 B_F 를 역DCT한다. 역DCT된 행렬을 B_C 라고 하면 다음과 같이 정리된다.

$$B_C = \begin{bmatrix} C^t & 0 \\ 0 & C^t \end{bmatrix} B_F \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} \quad (4)$$

C 행렬은 DCT변환 행렬로서 크기는 8×8 이다. 식 (4)에서 보는 바와 같이 B_c 는 화소 공간의 16×16 매크로블록이 되고, 화소 공간의 8×8 블록 B_R 은 행렬 R 을 적용하여 다음과 같이 얻을 수 있다.

$$B_R = RB_c R^t \quad (5)$$

R 행렬의 크기는 8×16 이고 다음과 같이 정의된다.

$$R = \frac{1}{2} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1 & 1 \end{bmatrix} \quad (6)$$

마지막으로 B_R 을 DCT 변환하면 8×8 DCT 공간의 행렬 B 가 생성된다.

$$B = CB_R C^t \quad (7)$$

위 식 (1)-(7)을 정리하면 T 변환행렬을 식 (8)과 같이 정의할 수 있다.

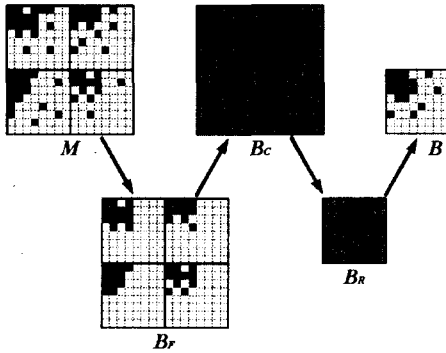


그림 1. DCT 공간에서의 매크로블록의 크기감소 과정

$$T = CR \begin{bmatrix} C^t & 0 \\ 0 & C^t \end{bmatrix} F \quad (8)$$

그림 1은 이러한 과정을 보여준다. 실선으로 표시된 블록은 화소 공간을 나타내며, 점선으로 표시된 블록은 DCT 공간을 나타낸다. DCT 공간블록에서의 검은 점은 0이 아닌 DCT 계수값이고, 나머지 흰 점은 DCT 계수값이 0임을 뜻하며, 화소 공간블록에서의 회색 점은 화소값을 뜻한다.

변환행렬 T 는 미리 계산하여 완성된 하나의 행렬로 만들어 놓고 식 (1)을 바로 적용할 수 있다.

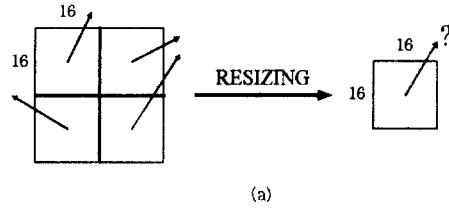
3. DCT 공간의 매체변환

2장에서 설명한 알고리즘을 실제 H.263 스트림에 적용시키려면 인트라프레임(Intraframe)에서는 아무 문제없이 사용할 수 있으나, 인터프레임(Interframe)에는 고려해야 할 사항이 많아진다.

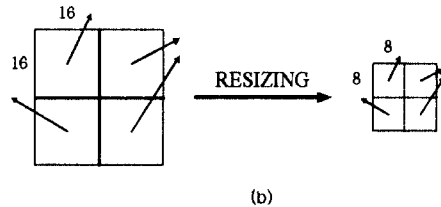
첫째, 현재 인터 프레임의 DCT 계수는 참조 프레임과의 움직임 벡터와 예측 오차(Prediction Error)로 인해 움직임 보상(Motion Compensation)되는 것이므로 두 정보중 하나라도 오류가 생기면 문제가 발생한다. 둘째, 인터 프레임에도 인트라 모드(Intra Mode)와 인터 모드(Inter Mode) 매크로블록이 존재하기 때문에 이 점 또한 고려해 주어야 한다.

첫 번째 문제가 발생하는 이유는 그림 2에서 쉽게 보여진다. 매크로블록 네 개가 크기감소 후 하나의 매크로블록이 되는데, 그림 2(a)에서와 같이 각각의 매크로블록이 다른 움직임 벡터를 가지고 있다면, 크기감소 후에는 대표되는 하나의 움직임 벡터를 산출하기가 쉽지 않다. 이 문제를 해결하기 위해서 본 논문에서는 그림 2(b)와 같이 H.263에서 사용되는 향상된 예측 모드(Advanced Prediction Mode)를 도입했다. 이 모드는 하나의 매크로블록에 존재하는 네 개의 블록들이 각각 움직임 벡터를 가질 수 있으므로 움직임 벡터의 크기를 반으로 줄이는 것으로 오류를 막을 수 있다. 그러나, 움직임 벡터가 홀수 값을 가질 경우, 이 값을 반으로 줄일 때 반올림을 하는 것에 대한 약간의 오류가 발생한다. 만약 입력 스트림이 움직임이 거의 없는 영상일 경우는 거의 대부분의 매크로블록 모드가 인트라 모드가 되고, 이 작은 오류는 인트라 모드에 의해 갱신되지 않으므로, 계속 누적되어 전파되게 된다. 그래서 초기 부호화시에 강제로 인트라 매크로블록을 삽입하거나 강제로 인트라 프레임을 삽입하면서 부호화 된 스트림이라면 오류의 전파는 중도에 차단되게 된다. 상대적으로 움직임이 많은 스트림이나 화면전환이 많은 스트림의 경우에는 계속 인트라 모드 매크로블록이 생성되므로 오류전파를 막을 수 있다.

크로블록에 존재하는 네 개의 블록들이 각각 움직임 벡터를 가질 수 있으므로 움직임 벡터의 크기를 반으로 줄이는 것으로 오류를 막을 수 있다. 그러나, 움직임 벡터가 홀수 값을 가질 경우, 이 값을 반으로 줄일 때 반올림을 하는 것에 대한 약간의 오류가 발생한다. 만약 입력 스트림이 움직임이 거의 없는 영상일 경우는 거의 대부분의 매크로블록 모드가 인트라 모드가 되고, 이 작은 오류는 인트라 모드에 의해 갱신되지 않으므로, 계속 누적되어 전파되게 된다. 그래서 초기 부호화시에 강제로 인트라 매크로블록을 삽입하거나 강제로 인트라 프레임을 삽입하면서 부호화 된 스트림이라면 오류의 전파는 중도에 차단되게 된다. 상대적으로 움직임이 많은 스트림이나 화면전환이 많은 스트림의 경우에는 계속 인트라 모드 매크로블록이 생성되므로 오류전파를 막을 수 있다.



(a)



(b)

그림 2. 네 개의 매크로블록의 움직임 벡터 처리법: (a) 하나의 움직임 벡터로 대표되는 경우 (b) 향상된 예측 모드를 사용하는 경우

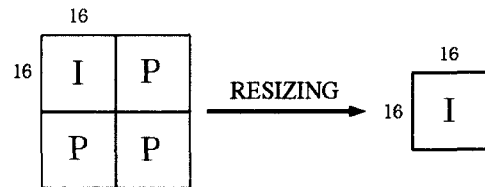


그림 3. 네 개의 매크로블록의 모드가 각각 다를 때 대표 모드 정하는 법

두 번째 문제 역시 고려해보면, 앞에서 제기되었던 것과 같이, 이것은 네 개의 매크로블록에서 다른 모드가 하나라도 들어 있을 때의 처리법이다. 그림 3에서 어떤 방법으로 대표 모드를 결정하는지를 보였다. 인트라 모드 매크로블록을 인터 모드 매크로블록으로 만들 수는 없지만, 역은 가능하다는 사실을 이용한다. 즉, 네 개의 매크로블록 중에서 인트라 모드 매크로블록이 하나라도 존재하면 인트라 모드로 대표되며 이것은 원래 영상에서 네 개의 매크로블록만을 움직임 보상으로 다시 DCT 한 후 2장에서 다룬 알고리즘을 사용함으로써 이루어진다.

이러한 방식으로 DCT 계수, 움직임 벡터 그리고 매크로블록 모드 정보를 갱신한 후에는 가변장 길이 부호화(Variable Length Coding) 모듈에 이 정보들을 넘겨주고 향상된 예측 모드로 재부호화 하면 크기감소된 스트림이 만들어진다.

4. 실험 결과

움직임이 많은 비디오영상인 Glasgow와 움직임이 적은 비디오 영상인 Akiyo, 또 카메라 회전 동작이 많은 Foreman에 적용시켜 보았다. 제안된 매체변환기의 입력으로 CIF 크기의 영상이 들어가게 되고, 출력으로는 QCIF 크기의 영상이 생성된다.

이것과 비교하기 위하여 QCIF 크기의 동일한 비디오영상의 타겟 비트율(Target Bit Rate)을 1/4로 줄인 상태로 부호화한 스트림과 비교해 보았다. 표 1에서는 각 실험영상에 대해 QCIF 크기로 부호화한 스트림을 복호화했을 때의 평균 PSNR과 CIF 크기로 부호화한 스트림이 제안된 알고리즘을 거친 후 생성된 스트림을 복호화했을 때의 평균 PSNR을 비교하였다.

표 1. 각 실험영상들의 평균 PSNR 비교

	QCIF 비디오영상 (ES)	CIF를 크기감소 시킨 비디오영상		
		모드 없음(NM)	강제 인트라 프레임 삽입(FF)	강제 인트라 매크로블록 삽입(FM)
Glasgow	25.268dB	27.960dB	30.269dB	27.958dB
Akiyo	33.814dB	33.325dB	35.978dB	38.210dB
Foreman	27.852dB	27.399dB	31.051dB	30.574dB

표 2는 각 실험영상에 대해 완전 복호화, 부호화가 일어나는 매체변환기와 본 논문에서 제안한 매체변환기의 각 모듈별 계산량을 나타낸다. 움직임 예측(ME), 움직임 보상(MC)과 역DCT과정이 없는 대신 제안된 알고리즘(DDR)의 연산 개수가 추가되었으나 연산량이 상당히 줄어든 것을 알 수 있다.

표 2. 각 실험영상들의 연산량 비교

	완전부호화/복호화 매체변환기				제안된 매체변환기		
	IDCT	DCT	ME	MC	IDCT	DCT	DDR
Glasgow	27796	6949	4984	19938	8732	8732	19064
Akiyo	13396	3349	7496	29987	0	0	13396
Foreman	33424	8356	4849	19396	3192	3192	30232

각 프레임에서의 PSNR을 그림 4-5에서 보였다. Akiyo 비디오 영상에서 강제 모드가 없는 스트림은 오류 전파로 인한 PSNR 감소가 보이지만 이는 강제 모드를 사용하면 막을 수 있다.

그림 6-7에서는 시각적인 화질을 비교해 보았다. 오른쪽 영상들은 어떠한 강제 모드도 포함하지 않은 영상이고, 시각적인 실험영상은 제안된 알고리즘의 오류 전파 정도를 알아보기 위해 20번째 프레임 이후에 복호화된 임의의 프레임을 선택하였다.

5. 결론

본 논문에서는 가변적인 대역폭을 가진 네트워크에 비디오를 전송하는데 있어서 비트율을 제어하는데 사용되는 매체변환기에 적용될 크기감소 알고리즘을 DCT 공간에서 수행하였다. DCT 계수와 이에 따른 변환 행렬을 이용하여 역DCT 과정 없이 프레임단위의 크기감소가 가능했으며, 이를 모든 프레임에 적용하면서 움직임 벡터와 그 밖의 움직임 정보, 매크로블록의 타입을 재이용하는 방법을 제안하였다. 이로 인해 완전 복호화와 재부호화에 따른 계산량을 줄임으로써 실시간 비트율 제어를 실현할 수 있다. 실험결과에서 알 수 있듯이 PSNR과 시각적 화질 또한 만족할만 하였다. 그러나 앞서도 언급되었던 움직임 벡터의 크기감소 정수화로 인해 정확한 위치에서 움직임 보상이 이루어지지 않을 수도 있는 문제가 있으며, 이에 의한 오류의 전파를 없애기 위한 방법을 역시 DCT 공간에서 수행할 수 있는 방법을 연구 중이다.

참고 문헌

[1] Navin Chaddha and Anoop Gupta, "A Frame-work for Live Multicast of Video Streams over the Internet", *IEEE*

International Conference on Image Processing, vol.1, pp.1-4, 1996
 [2] Oliver Werner, "Requantization for Trascoding of MPEG-2 Intraframes", *IEEE Trans. on Image Processing*, vol.8, no.2, Feb. 1999
 [3] Tamer Shanableh and Mohammed Ghanbari, "Heterogeneous Video Transcoding to Lower Spatio-Temporal Resolutions and Different Encoding Formats", *IEEE Trans. on Multimedia*, vol.2, no.2, pp.101-110, June 2000
 [4] Jeongnam Youn, Min-Ting Sun and Chia-Wen Lin, "Motion Vector Refinement for High-Performance Transcoding", *IEEE Trans. on Multimedia*, vol.1, no.1, pp.30-40, Mar. 1999
 [5] Changhoon Yim and Michael A. Isnardi, "An Efficient Method for DCT-Domain Image Resizing with Mixed Field/Frame-Mode Macroblocks", *IEEE Trans. on Circuits and Systems for Video Technology*, vol.9, no.5, pp.696-700, Aug. 1999

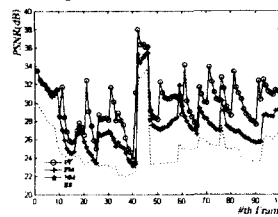


그림 4. Glasgow 비디오영상의 각 프레임에서의 PSNR

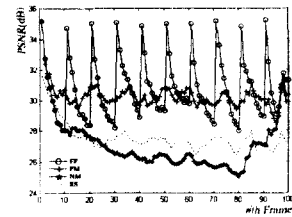
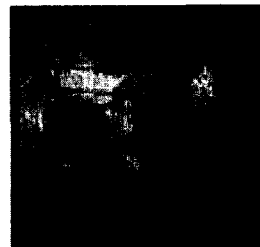
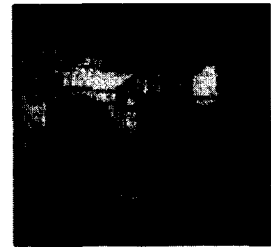


그림 5. Foreman 비디오영상의 각 프레임에서의 PSNR



(a)



(b)

그림 6. Glasgow 실험영상에 대한 시각적 화질비교: (a) QCIF 영상을 32kbps로 부호화한 경우 (b) CIF 영상을 128kbps로 부호화 후 QCIF로 크기감소한 경우



(a)



(b)

그림 7. Foreman 실험영상에 대한 시각적 화질비교: (a) QCIF 영상을 32kbps로 부호화한 경우 (b) CIF 영상을 128kbps로 부호화 후 QCIF로 크기감소한 경우