

압축된 영역에서의 영상크기 축소 트랜스코더의 성능 향상 방법의 연구

김윤형, 이조원, 임영환
승실대학교 컴퓨터학과
e-mail : yhkim602@media.ssu.ac.kr

A Study on Performance Improvement of a Downscaling Method for a Resolution Transcoder in a Compressed Domain

Yun Hyung Kim, Jo Won Lee, Young Hwan Lim
Dept. of Computer Science, Soongsil University

요 약

모바일 환경등의 저대역 통신망에서 VOD 서비스를 제공하기 위해서는 영상 데이터의 크기를 줄이는 방법이 요구된다. 데이터를 줄이는 방법 중 가장 효율적인 방법이 영상 크기 축소 트랜스코더(Resolution Downscale Transcoder)이지만 시스템의 리소스를 많이 소모하게 되므로 이를 줄이는 방법으로 움직임 예측을 최소화하는 움직임 벡터 병합 방법이 연구되었다. 하지만 이러한 방법들은 보다 정확히 움직임 벡터를 생성하지만 실제 영상에는 크게 적용되지 못하고 있다. 따라서 본 논문에서는 움직임 벡터를 병합하는 비율을 높여 움직임 예측을 줄이고 인트라 영역에서의 변환 과정을 간소화하여 전체적인 시스템 리소스의 소모를 줄이도록 하였다.

1. 서론

인터넷과 컴퓨터 시스템의 발달로 인하여 현대에는 VOD(Video On Demand)서비스가 대중화 되었고 모바일(Mobile)시스템의 발달로 모바일을 통한 VOD 서비스 또한 개시될 예정이다. 하지만 VOD 서비스에서 사용될 데이터들은 높은 대역폭을 요구하므로 모바일 환경에서 사용하기 위해 대역폭을 낮추는 방법이 필요하게 된다.

이 때문에 트랜스코더(Transcoder)가 필요하게 되는데 모바일 환경에서는 가장 큰 폭으로 대역폭을 줄일 수 있고 단말기에 작은 화면이 사용된다는 점에서 영상 크기 축소 트랜스코더(Resolution Downscaled Transcoder)가 가장 적절한 선택이라고 할 수 있다. 그러나 영상 크기 축소 트랜스코더는 다른 트랜스코더에 비해 시스템 리소스를 크게 요구하게 되므로 VOD 서비스 서버(Server)가 사용자의 요구를 원활히 응답할 수 없는 경우가 발생하게 된다. 따라서 영상 크기 축소 트랜스코더에 대한 연구는 시스템 리소스를 줄이는데 그 목적을 두고 진행되고 있으며 특히 가장 큰 리소스를 소모하는 움직임 예측(Motion Estimation)을 줄여 전체적인 성능향상을 높이는 데에 중점을 두고 있다.

움직임 예측에 대한 기존의 방법들은 변환될 대상 매크로블록이 모두 인트라 형식일 경우에 대해서 움직임 예측 없이 움직임 벡터를 생성하도록 하고 있다. 여기에

변환될 4 개의 대상 매크로블록 중 하나가 인트라 형식이거나 스킵 형식일 경우 또한 움직임 예측이 필요하지 않도록 한다면 보다 많은 움직임 예측 횟수를 줄일 수 있다. 또한 인트라 형식에서 변환에 사용되는 역 DCT 과정과 영상크기 축소(Downscale)과정을 간소화하여 개선하게 되면 상당한 비용 개선을 기대할 수 있을 것이다.

본 논문의 목표는 4 개의 매크로블록 중 하나가 인트라 형식이거나 스킵 형식의 매크로블록일 경우 기존의 방법에서 인트라 형식으로 변환되던 것을 인트라 형식으로 변환하여 움직임 예측이 사용되는 경우를 줄이는 방안과 DCT 영역에서 역 DCT 를 최소한 사용하고 영상 크기 축소 과정을 동시에 진행하는 방안을 제안하여 전체적인 비용적 측면에서의 성능 개선을 도모하는 방법을 제안하고 구현하는 것이다.

압축된 영역에서의 작업이기 때문에 영상의 화질은 제안된 방법을 사용하지 않은 경우와 다를 수 있다. 따라서 사용자의 최소한의 QoS 를 만족하는 범위에서 접근하여야 한다.

본 논문은 제 2 절에서 간소화된 인트라 영역에서의 변환 방법을 설명하였고 제 3 절에서 향상된 움직임 벡터 병합 방법을 설명하였다. 제 4 절에서는 여러 지표를 사용하여 본 논문에서 제안한 방법을 사용한 경우와 그렇지 않은 경우를 비교 평가하고 제 5 절에서 결론을 다루고 있다.

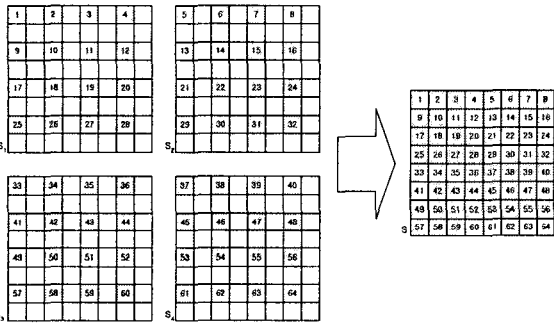
2. 인트라 매크로블록의 축소 방법

2.1 선택적 IDCT의 개요

인트라 영역에서의 변환 과정은 역 DCT(Inverse Discrete Cosine Transform)와 DCT(Discrete Cosine Transform) 그리고 영상 크기 축소(Downscale)과정이라 할 수 있는데 이들 과정 중 역 DCT와 영상 크기 축소 과정을 병합하여 진행하는 방법을 제안한다.

인트라 영역에서의 영상 축소 병합에 대한 알고리즘은 본 논문에서는 선택적 IDCT>Selective Inverse DCT)라는 방법을 제안한다.

이 알고리즘은 기존의 방법에서 4개의 매크로블록에 적용되는 역 DCT의 일부분을 수행하여 영상 크기 축소 방법으로 사용되는 방법 중 대표값 방법을 사용하는 것과 같은 결과를 얻으며 부분 역 DCT를 수행하기 때문에 기존의 방법에 비해 4분의 1의 역 DCT만을 수행하게 되며 영상 크기 축소 과정을 동시에 진행하는 효과를 가져오기 때문에 별도의 영상 크기 축소 과정을 거칠 필요가 없으므로 IDCT와 Downscale에 소모된 비용의 70% 정도를 줄일 수 있을 것이라 예상된다.



[그림 2-1] 선택적 IDCT

2.2 선택적 IDCT 알고리즘

선택적 IDCT 알고리즘은 다음과 같다.

- (1) 병합할 대상 DCT 블록 4개를 가져온다. 가져온 블록을 각각 D1, D2, D3, D4라 지칭한다.
- (2) D1, D2, D3, D4에 의해 생성되는 Sample Value의 블록을 S1, S2, S3, S4라 지칭하고, 병합될 Sample Value 블록을 S라 하고 이를 사용하여 생성될 DCT 블록을 D라 지칭한다.
- (3) 다음의 함수를 사용하여 블록 S를 생성한다.

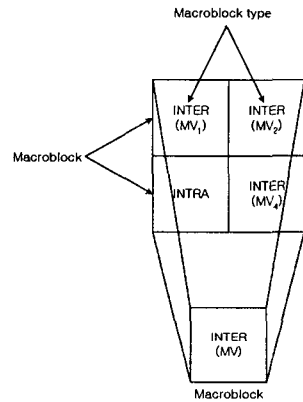
- a) $y < 4$ and $x < 4$ 일 경우
 $S(y, x) = S0(m, n)$
 $m = y \times 2, n = x \times 2$
- b) $y < 4$ and $x > 3$ 일 경우
 $S(y, x) = S1(m, n)$
 $m = y \times 2, n = (x - 4) \times 2$
- c) $y > 3$ and $x < 4$ 일 경우
 $S(y, x) = S2(m, n)$
 $m = (y - 4) \times 2, n = x \times 2$
- d) $y > 3$ and $x > 3$ 일 경우
 $S(y, x) = S0(m, n)$
 $m = (y - 4) \times 2, n = (x - 4) \times 2$

- (4) 생성된 Sample Value 블록인 S를 사용하여 FDCT를 통하여 하나의 DCT 블록 D를 생성한다.

3. 적극적인 움직임 벡터 병합 방법

움직임 예측(Motion Estimation)을 줄이기 위해서는 변환할 대상 매크로블록(Macroblock)의 형식이 동일하게 만들어져야 한다. 즉, 일부가 인트라 형식(Intra Type)이거나 스킵 형식(Skip Type)일 경우 이런 매크로블록을 움직임 벡터(Motion Vector)를 가진 인트라 형식(Inter Type)의 매크로블록으로 만들면 된다는 것이다.

본 논문은 대상 매크로블록이 여러 형식일 경우 대상 매크로블록 중 3개 이상이 인트라 형식의 경우에 대해 제안하고자 하는데 이 경우 기존의 방법론들이 변환될 4개의 대상 매크로블록이 모두 인트라 형식인 경우에 대한 적용 가능한 것에 비해 보다 많은 경우에 대해 적용 가능하다는 장점과 기존의 방법에 비하여 정확성이 감소하여 화질이 저하가 올 수 있다는 단점이 예상된다.



[그림 3-1] 움직임 벡터 추측 방법을 적용하는 경우

[그림 3-1]에서와 같은 경우 또는 3개의 인트라 형식 매크로블록과 하나의 스킵 형식 매크로블록이 대상 매크로블록일 경우 인트라 형식의 매크로블록이 아닌 매크로블록은 움직임 벡터가 존재하지 않기 때문에 이에 대한 움직임 벡터를 찾을 필요가 제기된다. 그러나 움직임 추출을 통하여 움직임 벡터를 찾을 경우 그 비용이 크므로 나머지 3개의 인트라 형식 매크로블록에 존재하는 움직임 벡터를 이용하여 존재하지 않는 움직임 벡터 mv_x 를 추측하고자 한다. 이때 추측된 움직임 벡터 값이 움직임 벡터 병합 시에 큰 영향을 주지 않게 하기 위해 다음의 수식에서와 같이 나머지 움직임 벡터의 값 중 최소의 값을 선택하게 하였다.

$$\overline{mv}_x = \min(\overline{mv}_1, \dots, \overline{mv}_4) \quad \text{수식 1}$$

위의 수식으로 움직임 벡터를 생성하면 필요한 움직임 벡터 4개가 준비되며 이들 움직임 벡터를 다음 수식에 적용하여 변환 시 움직임 추출을 거치지 않고 새로운 움직임 벡터 mv' 를 생성하게 하여 움직임 추출에서 발생하는 비용을 줄이게 된다.

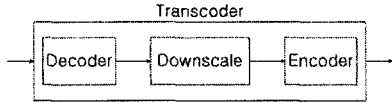
$$\overline{mv}' = \frac{1}{2} \frac{\sum_{i=1}^4 \overline{mv}_i w_i}{\sum_{i=1}^4 w_i} \quad \text{수식 2}$$

수식 2에서 w_i 는 가중치로 움직임이 큰 움직임 벡터가 새로운 움직임 벡터값에 보다 크게 반영되게 하기 위하여 벡터의 스칼라 값을 가중치로 이용하였다.

$$w_i = \sqrt{x_i^2 + y_i^2} \quad \text{수식 3}$$

4. 시스템 구조 및 성능 평가

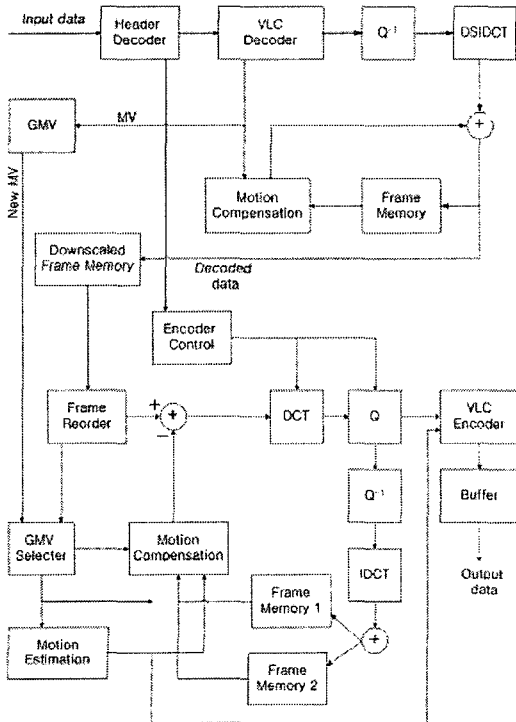
4.1 전체 시스템 구조



[그림 4-1] 일반적인 영상 크기 축소 트랜스코더 시스템 구성도

일반적인 영상 크기 축소 트랜스코더(Resolution Downscaled Transcoder)는 [그림 4-1]과 같이 3개의 부분으로 구성되며 본 논문에서는 [그림 4-2]의 시스템으로 구성된다.

복호기(Decoder)에서 MPEG 데이터를 읽어 헤더를 분석하여 부호기(Encoder)에 필요한 정보(비트율, 가로, 세로 크기, GOP 구성 등)들을 넘기며 VLC Decoder에서 움직임 벡터(Motion Vector)를 추출하여 GMV 부분으로 넘긴다. GMV는 적용 가능한 매크로 블록들에 대하여 Guessing Motion Vector Method를 이용하여 새로운 움직임 벡터를 생성하며, 부호기 부분에서 생성된 움직임 벡터에 해당하는 매크로블록을 처리할 때 움직임 예측(Motion Estimation)을 거치지 않고 GMV 부분에서 생성한 움직임 벡터 값을 사용하게 된다. 복호기 부분의 DSIDCT 부분은 선택적 IDCT를 처리하는 부분이다.



[그림 4-2] 영상 크기 축소 트랜스코더의 전체 시스템 구성도

4.2 성능평가

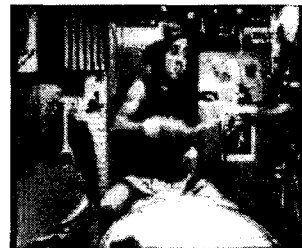
성능 평가 항목으로 압축률을 나타내는 비트율(Bitrate)과 화질 손실도 PSNR(Peak Signal to Noise Ratio)와 실제 변환 과정에서 움직임 예측 정도를 비교하였으며 사용된 영상 데이터는 두 가지이며 다음의 특성을 갖는다.

- Sample 1 - 원본 크기 : 576 × 320 Pixel
- 변환 후 크기 : 288 × 160 Pixel
- 속도 : 25 FPS
- GOP 구조 : IBBPBBPBBPBBPBB
- 프레임 수 : 15 개
- 원본의 비트율 : 1200 Kbits/sec



[그림 4-3] "Sample 1"의 한 장면

- Sample 2 - 원본 크기 : 352 × 288 Pixel
- 변환 후 크기 : 176 × 144 Pixel
- 속도 : 25 FPS
- GOP 구조 : IBBPBBPBBPBBPBB
- 프레임 수 : 12 개
- 원본의 비트율 : 1200 Kbits/sec



[그림 4-4] "Sample 2"의 한 장면

4.2.1 비트율(Bitrate) 평가

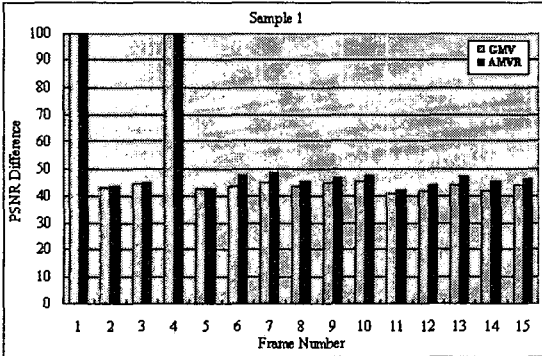
원본 영상들을 영상 크기 축소 변환 부호화를 하는 목적은 원본 영상 비트율의 4분의 1의 비트율을 가지게 영상을 변환시키는 것으로 [표 4-1]에서와 같이 복호화와 영상크기 축소, 부호화를 단계적으로 처리한 데이터의 비트율은 원본의 약 4분의 1이 되며 GMV 알고리즘을 적용시킨 영상 크기 축소 트랜스코더가 변환한 데이터 역시 전체처리보다는 조금 크지만 약 4분의 1임을 알 수 있다. 즉 비트율을 낮추는 데에는 성공적이다.

	GMV	Full Processing
Sample 1	288 Kbits/s	285Kbits/s
Sample 2	294 Kbits/s	291Kbits/s

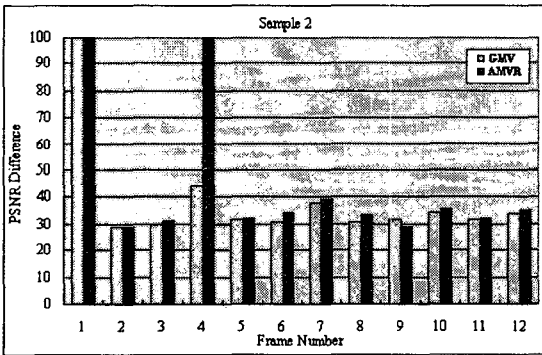
[표 4-1] 비트율 비교표

4.2.2 화질평가

화질을 분석적인 방법으로 평가하기 위하여 원본 이미지와 대상이미지와의 차이를 비교하여 상대적으로 손상 정도를 평가하는 PSNR(Peak Signal to Noise Ratio) 방식을 사용하였다. 원본 이미지는 전체 처리로 생성된 데이터를 사용하였고 이미지의 밝기 정보(Luminance Channel)만을 사용하였으며 비교자료로써 AMVR 을 적용시켰을 때의 값도 표시하였다.



[그림 4-5] "Sample 1"의 PSNR(Luminance Channel)



[그림 4-6] "Sample 2"의 PSNR(Luminance Channel)

[그림 4-5]와 [그림 4-6]에서와 같이 영상의 압축 순서 (I→ P→ B→ B, 즉 그림에서 1→ 4→ 2→ 3)에 따라 진행될수록 손상이 심해짐을 볼 수 있지만 AMVR 과 비교시 크게 떨어지지 않음을 알 수 있다.

주관적인 방법으로 ITU 권고안(Recommendation) 500-5 에 정의된 관찰자 평가 방식으로 평가 시 대체적으로 4 등급으로 평가된다.([표 4-2] 참조)

등급	시각적 손상
5	극히 적음
4	약간 인식, 하지만 크게 손상되지 않음
3	손상이 약간 인식
2	손상
1	매우 손상

[표 4-2] 시간적 화질 등급 [3]

4.2.3 움직임 벡터 병합 효율

본 논문에서 제안된 GMV 를 적용할 경우 부호화 부분에서 움직임 예측을 하지 않고 생성한 움직임 벡터를 이용하여 비율을 측정하였다. 이때 1 프레임은 측정치에 포함되지 않았으며 비교 대상으로 4 개의 움직임 벡터가 대상일 경우만 병합하는 기존의 방법론을 적용시켰을 경우의 효율을 같이 표시하였다.

[표 4-3]에서와 같이 제안된 GMV 방안은 기존의 방법론의 2 배 가까이 움직임 예측을 하지 않는 효율을 보여준다. 즉, 처리 속도 역시 기존의 방법론보다 증가될 것이라 기대할 수 있다.

	GMV	기존 방법론	전체처리
MV 병합으로 처리된 MB	52%	28%	0%
움직임 예측으로 처리된 MB	48%	72%	100%

[표 4-3] 움직임 벡터 병합 효율

5. 결론

저대역 통신망에서 VOD 서비스를 제공하기 위해서는 데이터의 크기를 최소화하는 방법이 요구된다. 압축된 영상 데이터의 크기를 줄이는 방법들 중 가장 효율적인 방법이 영상 크기 축소 트랜스코더(Resolution Downscale Transcoder)이지만 이 방법은 많은 시스템 리소스를 소모하므로 이를 줄이는 방법으로 움직임 예측(Motion Estimation)을 최소화하는 방법인 움직임 벡터(Motion Vector) 병합 방법이 연구되었지만 이러한 방법들은 정확히 새로운 움직임 벡터를 생성하지만 실제 영상에서는 크게 적용되지 못하고 있다.

본 논문에서 제시된 방법은 화질면에서는 기존의 방법들에 비해 손상이 있으나 움직임 예측 횟수는 기존의 방법보다 대폭 줄일 수 있었으며 인트라 영역에서의 변환 과정 또한 간소화하여 전체적인 시스템 리소스 소모를 줄이는 데 성공하였다. 따라서 모바일(Mobile)등의 저대역 통신망에서 VOD 서비스를 제공하는 서버에 대한 부담을 크게 줄일 수 있을 것이라 기대된다.

6. 참고문헌

- [1] Niklas Bjork and Charilaos Christotoulos, "Transcoder Architectures for Video Coding," IEEE Transactions on Consumer Electronics, Vol. 44, No. 1, February 1998.
- [2] Bo Shen, Ishwar K. Sethi, and Bhaskran Vasudev, "Adaptive Motion-Vector Resampling for Compressed Video Downscaling," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 9, No. 6, September 1999.
- [3] Martyn J.Riley and Iain E.G.Richardson, "Digital Video Communications," Artech House, 1997.
- [4] ISO/IEC 13818-2 MPEG2, "Generic coding of moving pictures and audio," Nov., 1993.
- [5] MSSG, "MPEG-2 Source Code," 1994.
- [6] M. R. Hashemi, L.Winger, and S. Panchanathan, "Compressed Domain Motion Vector Resampling for Downscaling of MPEG Video".