

DCT 임베디드 동영상 부호화기의 비트율 제어

조정식, 이연문, 정차근
호서대학교 제어계측공학과
041-540-5429 / 041-540-5480

Rate Control for DCT Based Embedded Video Coder

Kyungsik Cho, Youn-Moon Lee, Cha-Keon Cheong
Dept. of C&I Eng., College of Eng., Hoseo University
e-mail : 99140301@icando.hoseo.ac.kr

요 약

본 논문에서 제안한 방법은 DCT 임베디드 동영상 부호화기를 사용하여 부호화기의 레이트 디스토션 성능과 기존 프레임과 예측 프레임간의 의존성을 이용한 디스토션이 일정한 효율적인 비트율 제어 알고리즘을 제안한다. 다양한 표준 동영상에 대해 컴퓨터 모의 실험을 수행하고 기존 방법과의 비교를 통해 제안방법의 유효성을 검증하고 제안된 알고리즘의 부호화 효율을 확인했다.

1. 서론

정보통신과 인터넷의 발달로 유무선 환경에서의 효율적인 동영상 부호화가 필요하게 되었다. 이에 따라 H.26x, MPEG 등 동영상 부호화기의 국제표준화가 진전되어 이들의 실용화 및 상업화가 급속히 진행되고 있으며 많은 연구가 진행되고 있다. 특히 IMT2000과 같은 무선환경에서는 전송대역폭이 좁아 영상신호의 전송에는 비트 레이트가 많은 제약이 받는다. 이로 인하여 블록 노이즈의 발생과 함께 수신 영상의 품질이 저하되는 문제점을 가지고 있다. 이와 같이 낮은 비트 레이트에서의 동영상 부호화에서는 제한된 비트를 효율적으로 사용하기 위해서 비트율 제어가 필수적이다.

기존 레이트 제어 방법 중 MPEG의 TM5의 경우 레이트 제어를 위해 버퍼에 따른 양자화 크기를 조정한다. 이 방법은 구조는 간단하지만 디스토션에 대해 최적화 되어 있지 않아 복호화

된 동영상의 화질이 떨어지며 프레임 skip이 발생하는 단점을 가지고 있다. 따라서 본 논문에서는 정확한 레이트 제어가 가능한 임베디드 영상 부호화기를 사용한 DCT 임베디드 동영상 부호화기를 사용하여 레이트 제어방법을 적용하였다. 본 논문에서 제안한 레이트 제어 방법은 각 프레임에서의 부호화 효율과 기존 프레임과 예측 프레임간의 의존성을 이용하여 디스토션 값을 일정하게 두고 각 프레임의 비트량을 제어하는 방식이다. 이 방법은 기존의 방법에 비해 디스토션이 일정한 특성을 가지며 프레임 skip이 발생하지 않은 좋은 성능을 나타냈다.

본 논문의 구성은 2장에서 간단히 DCT 임베디드 동영상 부호화기에 대하여 논하였고, 3장에서는 부호화 효율 관계를 나타내는 파라미터 β 와 프레임간의 의존관계를 나타내는 파라미터 α 에 대해 다룬 후 이를 바탕으로 디스토션이 일정한 레이트 제어 방법에 대하여 기술하였다. 4장에서는 표준 동영상에 대해 컴퓨터 모의 실험을 수행하고 기존 방법과의 비교를 위해 기존 H.263의 TM5[1]과 비교를 통해 제안방법의 유효성을 검증하고, 제안된 알고리즘의 부호화 효율을 확인했다.

2. DCT 임베디드 동영상 부호화기

본 논문에서 사용한 동영상 부호기는 기존 방식의 양자화 방식 대신에 계수들을 제로 트리 구조로 재배치한 후 임베디드 제로 트리 부호화의 양자화 방법인 연속 추정 양자화(SAQ)를 했다. 일반적으로 전 프레임에 대한 DCT 변환에 의한 화상의 표현은 웨이브렛 변환에 의한 화상의 표현과 동일하다[2]. 이 성질을 이용해서, DCT에 의한 입력 화상의 대역 분할을 피라미드 구조로 재배치함으로써, DCT의 높은 에너지 compaction 특성을

활용하여 부호화 성능을 개선시키는 것이 가능하다. 재배치 방법은 한 프레임 전체에 대해 8x8크기의 각 블록에 대한 DCT변환으로부터, DC성분, 저역 주파수성분, 고역 주파수 성분으로의 3레벨 피라미드 구조로 재배치한다. 재배열된 제로 트리 구조의 DCT계수를 양자화 하기 위해 향상된 임베디드 제로 트리 부호화의 한 방법으로 SPIHT[3]을 이용하였다. 그리고 부호화 효율을 높이기 위해 산술부호화[4]를 수행했다.

3. 레이트 제어 알고리즘

전송되는 비디오 스트림이 하나의 인트라 프레임(I)으로 시작하여 N-1개의 예측 프레임(P)으로 구성되어 전송된다고 가정하고 채널 폭이 C b/s이고 하나의 GOP의 경과 시간이 T초라면 각 GOP에 할당되는 레이트는 식(1)과 같다.

$$R_{GOP} = CT \quad (1)$$

동영상 부호화기에서 레이트 제어 문제는 식(2)에서와 같이 각 GOP내의 N개의 프레임에 대해 디스토션(D_i)을 최소로 하면서 $\text{bit}(R_i)$ 를 할당하는 것이다.

$$\begin{cases} R_1 + R_2 + R_3 + \dots + R_N = R_{GOP} \\ \min(D_1 + D_2 + D_3 + \dots + D_N) \end{cases} \quad (2)$$

기존 방법의 경우 제어 파라미터를 양자화 크기를 사용하였지만 본 논문에서는 임베디드 부호화기를 사용하였기 때문에 제어 파라미터로서 비트 레이트 R_i 를 직접 사용할 수 있다.

식(2)을 최적화 하는 문제를 풀기 위한 방법으로 식(3)과 같은 Lagrangian방법[5]를 사용할 수 있지만 이 방법의 경우 디스토션의 변동폭이 크고 일정하지 않았으며 계산 량이 많았다. 그래서 본 논문에서는 디스토션 값을 일정하게 두고 레이트 값을 구하는 방법으로 접근하였다.

$$\begin{aligned} R(R_1, \dots, R_N) &= \sum_{i=1}^N D_i + \lambda (\sum_{i=1}^N R_i - R_{GOP}) \\ &= D_1 + D_2 + \dots + D_N \\ &\quad + \lambda (R_1 + R_2 + \dots + R_N - R_{GOP}) \end{aligned} \quad (3)$$

(a) 부호화 효율 파라미터 β

I와 P프레임에 대한 레이트 R에 대한 디스토션 D와의 관계는 식(4)과 같이 근사화 할 수 있다[5].

$$D = D_{\max} 2^{-BR} = \sigma^2 2^{-BR} \quad (4)$$

여기서 D_{\max} 는 부호화 레이트 R = 0일 때의 값이며 I 프레임의 경우 영상의 부호화기전의 분산치 σ^2 과 같고 P 프레임의 경우 움직임 보상 에러의 분산치와 같다. 부호화 효율 β 는 비트 레이트에 대한 디스토션의 감쇠 비를 나타내는 파라미터로 β 값

이 클수록 효율적인 부호화기이다. 실험 데이터를 통해 근사화하여 I와 P프레임에 대한 β 값을 구할 수 있다.

(b) 프레임 의존관계 파라미터 α

I 프레임의 화질이 좋을수록 움직임 보상 영상의 화질을 향상시켜 다음 P 프레임을 부호화 하는데 드는 비트 량을 줄여준다. 원형의 기준 프레임을 $g(i, j)$, 복호화된 기준 프레임을 $\hat{g}(i, j)$, 그리고 예측 프레임을 $f(i, j)$ 라고 놓으면, 기준 프레임에 대한 움직임 보상 에러는 식(5)과 같이 쓸 수 있다.

$$e(i, j) = f(i, j) - g[d(i, j)] \quad (5)$$

여기서 $d(i, j)$ 는 (i, j) 에서 움직임 벡터만큼 이동된 것을 의미한다. 움직임 보상에러의 분산치는 식(6)과 같다.

$$\begin{aligned} \sigma_e^2 &= E[e^2(i, j)] \\ &= E\{(f(i, j) - g[d(i, j)])^2\} \end{aligned} \quad (6)$$

실제 동영상 복호화 시에는 원형의 기준 프레임을 알 수 없기 때문에 실제 움직임 보상 에러는 식(7)과 같이 복호화된 영상 $\hat{g}(i, j)$ 로 부터 구해야 하며 이때의 분산치는 식(8)과 같다.

$$\hat{e}(i, j) = f(i, j) - \hat{g}[d(i, j)] \quad (7)$$

$$\begin{aligned} \hat{\sigma}_e^2 &= E[\hat{e}^2(i, j)] \\ &= E\{(f(i, j) - \hat{g}[d(i, j)])^2\} \end{aligned} \quad (8)$$

이 때 실제 움직임 보상 에러 식(7)은 식(9)과 같이 쓸 수 있다.

$$\hat{e}(i, j) = \{f(i, j) - g[d(i, j)]\} + \{g[d(i, j)] - \hat{g}[d(i, j)]\} \quad (9)$$

우측항의 첫 번째 항은 원형 기준 프레임에 대한 움직임 보상 에러를 의미하며 두 번째 항은 움직임 변환된 기준 프레임의 에러를 나타낸다. 이 두 항이 독립이라면 식(9)으로부터 식(10)과 같이 쓸 수 있다.

$$\begin{aligned} \hat{\sigma}_e^2 &= \sigma_g^2 \\ &\quad + E\{(g[d(i, j)] - \hat{g}[d(i, j)])^2\} \end{aligned} \quad (10)$$

식(10)의 우측항의 두 번째 항은 원형 기준 프레임에 대한 디스토션을 의미하는게 아니라 움직임 변환된 기준 프레임에 대한 디스토션을 나타낸다. 이 두 디스토션의 관계는 선형관계를 가지며 식(11)과 같이 쓸 수 있다.

$$\begin{aligned} E\{(g[d(i, j)] - \hat{g}[d(i, j)])^2\} &\approx \alpha D_f \\ D_f &= E\{(g(i, j) - \hat{g}(i, j))^2\} \end{aligned} \quad (11)$$

식(11)에서 D_f 는 기준 프레임의 디스토션을 의미하며, (10)식과 (11)식으로부터

$$\hat{\sigma}_g^2 = \sigma_g^2 + \alpha D_f \quad (12)$$

여기서 α 를 프레임 의존 관계를 나타내는 파라미터로 사용하였다. 식(12)의 관계는 실험 데이터를 통해서 구할 수 있다. α 값의 범위는 0.5~0.9이며, 움직임이 거의 없는 경우 1에 가깝고 움직임이 많거나 화면 전환이 있는 경우 값이 작아진다[5].

(c) R_i 의 계산 방법

각 프레임의 디스토션 D_i 는 식(4)으로 부터 (13)식과 같이 쓸 수 있다.

$$D_i = \begin{cases} \sigma_1^2 2^{-\beta_1 R_i} & , \text{for } i=1 \\ \hat{\sigma}_i^2 2^{-\beta_i R_i} & , \text{for } i=2,3,\dots,N \end{cases} \quad (13)$$

여기서 σ_1^2 은 원영상의 분산치를 $\hat{\sigma}_i^2$ 은 움직임 보상 에러의 분산치를 나타낸다. $\hat{\sigma}_i^2$ 값은 식(12)로부터 다음과 같이 쓸 수 있다.

$$\hat{\sigma}_i^2 = \sigma_i^2 + \alpha_i D_{i-1} \quad (14)$$

(14)식을 (13)식에 넣어 R_i 에 대하여 수식을 전개하면 (15)식이 나온다.

$$R_i = \begin{cases} \frac{1}{\beta_1} \log_2 \frac{\sigma_1^2}{D_1} & , i=1 \\ \frac{1}{\beta_i} \log_2 \frac{\alpha_i D_{i-1} + \sigma_i^2}{D_i} & , i=2,3,\dots,N. \end{cases} \quad (15)$$

$D_i, \alpha_i, \beta_i, \sigma_i^2$ 값이 주어지면 식(15)으로 부터 R_i 값을 구할 수 있다. 여기서 디스토션이 일정하게 하기 위해 D_i 값은 일정하게 두었다. 이 때 동영상의 경우 인트라 프레임의 부호화 효율은 인트라 프레임에 많은 영향을 주기 때문에 인트라 프레임 디스토션 ($D_I = D_1$)와 인트라 프레임 디스토션 (D_P)을 같게 두지 않고 다음과 식과 같이 D_I 를 D_P 보다 작게 하여 부호화 효율을 높일 수 있다.

$$D_I = k D_P \quad (0 < k < 1) \quad (16)$$

그러면 R_GOP를 만족하는 ($D_g = D_p$)값은 다음 알고리즘에 의해 결정된다.

1. $R(D_{g1}) < R_{GOP} < R(D_{g2})$ 가 되도록 D_{g1}, D_{g2} 초기화.
2. $D_g = (D_{g1} + D_{g2})$ 라 놓고,
if $R(D_g) < R_{GOP}$ then
replace D_{g1} with D_g

else

replace D_{g2} with D_g

3. Repeat step 2 until $|R(D_g) - R_{GOP}| < \epsilon$,

ϵ 은 0에 가까운 양수

$R(D_g)$ 는 디스토션이 D_g 일 때의 GOP의 총 비트 량.

D_I, D_P 의 관계를 나타내는 k값은 실험에 의해 0.8정도가 적당하였다. 그림 (1)은 k=0.8일 때의 결과와 D_I, D_P 값이 같을 때를 비교한 그래프로 PSNR이 평균 0.1dB 향상되었다.

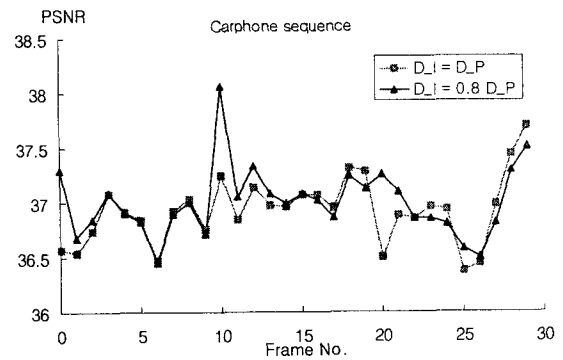


그림 1 D_I, D_P 에 따른 PSNR 그래프

4. 실험방법 및 결과

176x144 QCIF foreman, carphone 휘도 영상 30 frames/s에 대하여 H.263 TM5와 제안한 방법을 비교하였다. GOP는 10 frame으로 하나의 I 프레임과 9개의 P 프레임으로 구성되며 bit rate를 256kb/s로 하였고 1초 동안 30 프레임에 대하여 실험하였다. 그림 (2)는 각 프레임에서의 PSNR값과 bit rate를 나타낸다. PSNR값을 보면 제안 방법이 H.263 TM5보다 변동폭이 적었으며 프레임 skip이 발생하지 않고 좋은 성능을 보였다.

5. 결론

본 실험에서는 DCT 임베디드 동영상 부호화기를 사용한 레이트 제어 방법을 제안하였다. 레이트 제어를 위해 정확한 레이트 제어가 가능한 임베디드 동영상 부호화기에 레이트 디스토션 특성과 프레임간의 의존성 파라미터를 사용하였다. GOP내 디스토션이 일정하다고 두고 각 프레임의 레이트를 조정하여 원하는 레이트가 되었을 때의 디스토션을 사용하여 디스토션이 일정하면서 성능이 우수한 부호화기를 얻을 수 있었다.

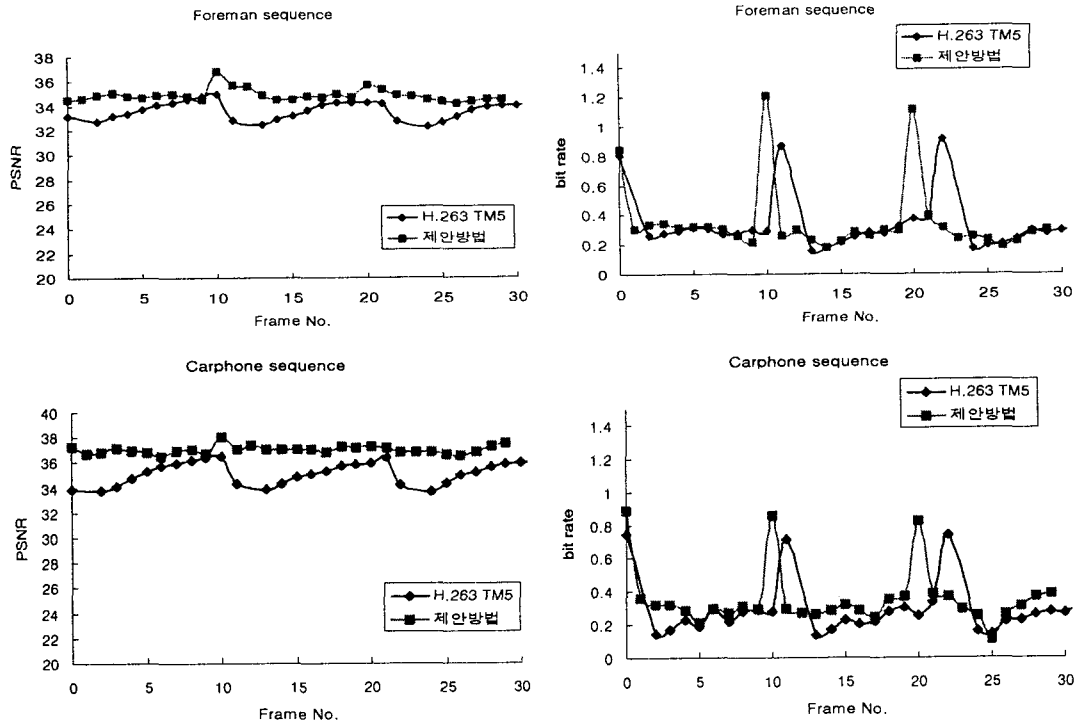


그림 2 프레임 번호에 따른 PSNR과 bit-rate

<참고문헌>

[1] DRAFT ITU-T Recommendation H.263
 [2] M. Ve Herli and C. Herley "Wavelets and filter banks: Theory and design," IEEE Trans. Signal Processing, Vol. 40, no. 9, pp. 2207-2232
 [3] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," IEEE Trans. Circuits and Systems for Video Technology, Vol. 6, No. 3, pp. 243-250, June 1996.
 [4] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," Commun. ACM. vol. 30, pp. 520-540, June 1987.
 [5] Po-Yuen Cheng, Jin Li, and C.-C Jay Kuo, "Rate Control for an Embedded Wavelet Video Coder." IEEE Trans. Circuits and Systems for Video Tech, Vol. 7, no. 4, pp.696-702, Aug. 1997
 [6] J. M. Shapiro, "Embedded image coding using zerotree of wavelet coefficients," IEEE Trans. on Signal Processing, Vol. 41, No. 12, pp 3445-3456, Dec 1993.
 [7] A. Islam and W. Pearlman, " An embedded and efficient low-complexity hierarchical image coder."
 [8] Z. Xiong, O. Guleryuz, and M. T. Orchard, "A DCT-based Embedded Imge Coder," IEEE Signal Processing Letters, vol. 3,

pp.289-290, Nov. 1996.

[9] W. B. Pennebarker and J. L. Mitchell, JPEG Still Image Data Compression Standard, New York: VanNostrand Reinhold, 1992.

[10] KHalid Sayood, Introduction to Data Compression, Morgan Kaufmann, 1996.