

# MPEG-4 비디오 부호화를 위한 객체기반 비트율 제어 방법

## Object-based Rate Control for MPEG-4 Video Coding

이정우, 호요성

광주과학기술원 정보통신공학과  
광주광역시 북구 오룡동 1번지

### 요약

본 논문은 MPEG-4 표준을 따르는 객체기반 비디오 부호화에서 고정 비트율을 지원하는 알고리즘을 기술한다. 각각 독립적으로 부호화되는 객체의 왜곡과 건너뛰어지는 객체의 왜곡을 계산하여 시간적-공간적으로 타당한 부호화 모드를 선택하는 방법과 버퍼를 제어하는 방법을 제안한다. 제안된 알고리즘은 낮은 비트율과 높은 비트율에서 모두 향상된 성능(PSNR)을 제공하며, 버퍼를 정확하게 제어하기 때문에 버퍼 넘침과 결핍 현상이 발생하지 않는다.

### 1. 서론

H.263, MPEG-1, MPEG-2, MPEG-4와 같은 비디오 부호화 표준들은 가변 길이 부호화(Variable Length Coding) 방법을 사용하기 때문에 비디오 부호화 과정을 통해 각 화면마다 발생하는 비트 수가 변한다. 이들 가변 비트율을 전송 대역폭이 고정된 채널을 통해 전송하고자 할 때, 보통 출력 비트율의 동적인 변동을 평활화하기 위해 채널 버퍼가 사용되며, 채널의 버퍼 넘침이나 결핍을 방지하기 위해 비트율 제어가 요구된다.

매크로블록의 양자화 변수값의 변화만을 통해 비트 발생량을 제어하는 MPEG-1, MPEG-2 표준과는 달리 H.263과 MPEG-4 표준에서는 비트 발생량을 제어할 수 있는 방법으로 화면의 건너뛴을 지원한다. 부호화 과정에서 건너뛰어진 화면은 복호화 과정에서 이전에 복원된 화면을 그대로 다시 보여주기 때문에, 공간적 화질 이외에 시간적 화질도 부호화 성능을 측정하는 중요한 성분이 된다.

현재까지 제안된 알고리즘들은 화면의 건너뛴이 단지 버퍼 상태에 따라 결정되기 때문에 복원 후에 재구성된 건너뛰어진 화면들은 화질의 열화를 초래하게 된다. 따라서 버퍼의 상태 뿐만 아니라 화면의 특성을 고려하여 부호화 할 수 있는 알고리즘이 요구된다. 또한 MPEG-4 비디오 부호화 알고리즘[1]은, 기존의 다른 표준과는 달리, 객체기반 부호화를 수행한다. 따라서 MPEG-4 비

트율 제어 알고리즘에서는 모양 정보를 부호화해서 발생하는 비트량도 중요하게 고려해야 한다.

최근에 A. Vetro에 의해 부호화 화면과 건너뛰어진 화면의 왜곡을 예측할 수 있는 알고리즘[2]과 비트율 제어 알고리즘[3]이 제안되었다. 이 방법은 공간적 화질과 시간적 화질 사이의 trade-off를 고려하여 공간적 화질을 낮추면서 더 많은 화면을 부호화할 것인지, 좀더 적은 수의 화면을 부호화하면서 공간적 화질을 높일 것인지를 결정한다. 하지만 이 알고리즘은 한 화면 전체나 한 개의 객체만을 부호화하기 때문에 다중 객체 부호화 알고리즘에 직접 적용하기 곤란하다. 또한 최적의 화면 건너뛴 비율(frame skip rate)을 계산하기 위해 정해진 최대값 내에 있는 모든 값에 대해 왜곡을 계산하기 때문에 많은 연산이 요구된다.

본 논문에서는 기존의 화면기반 왜곡예측 방법을 객체기반 왜곡예측 방법으로 확장하고, 객체기반 부호화 성능을 향상시키기 위한 새로운 비트율-왜곡(rate-distortion, R-D) 부호화 모드와 비트율 제어 알고리즘을 제안한다. 제안된 모드는 부호기에 의해 최적의 R-D 성능을 유지하도록 자동적으로 선택된다.

### 2. 비트율-왜곡 모델

영상 부호화 방식에서 부호화되는 화면의 왜곡은 다음과 같이 표현될 수 있다 [2].

$$D_c(Q_i) = a \cdot 2^{-2R'(I_i)} \cdot \sigma_i^2 \quad (1)$$

여기서  $\sigma_i^2$ 는 화면  $i$ 의 분산값이고,  $R'(I_i)$ 는 화면 내에 있는 화소당 평균 비트율,  $Q_i$ 는 양자화 변수,  $a$ 는 입력 신호와 양자화 특성에 의해 결정되는 상수이다.

복호기에서 건너뛰어진 화면을 복원할 때, 바로 이전에 부호화된  $i$ 번째 복원 화면을 그대로 사용하는 경우에,  $k$ 번째 건너뛰어진 화면의 왜곡은 다음과 같이 표현된다.

$$D_s(Q_i, k) = D_c(Q_i) + E\{\Delta^2 z_{i,k}\} \quad (2)$$

$$E\{\Delta^2 z_{i,k}\}$$

여기서  $t_i$  는  $i$  번째 화면과  $k$  번째 화면 사이의 예측 오류로써, optical flow를 이용하여 다음과 같이 정의된다.

$$E\{\Delta^2 z_{i,k}\} = \sigma_{x_i}^2 \sigma_{\Delta x,k}^2 + \sigma_{y_i}^2 \sigma_{\Delta y,k}^2 \quad (3)$$

여기서  $\sigma_{x_i}^2$ 와  $\sigma_{y_i}^2$ 는 각각  $i$  번째 화면에서  $x$ 와  $y$  방향으로 기울기 값들의 분산 값을 나타내고,  $\sigma_{\Delta x,k}^2$ 와  $\sigma_{\Delta y,k}^2$ 는  $x$ 와  $y$  방향으로 움직임 벡터들의 분산 값을 나타낸다. MPEG-4 다중 객체 부호화를 지원하기 위해서는 식 (1)과 식 (2)의 모든 값들은 객체가 정의되는 화소 내에서만 계산되어야 한다. 특히, 식 (1)에서 화소당 비트율은 모양 정보, 움직임 정보를 포함하는 헤더 정보를 고려하여 계산되어야 한다.

### 3. 비트율-왜곡 부호화 모드

영상 부호화기의 성능을 향상시키기 위해 R-D 부호화 모드를 이용하여 다음 화면을 부호화 하는 것이 좋은지, 건너뛰는 것이 좋은지를 결정한다. 이와 같이 제안된 알고리즘은 고정된 간격으로 화면을 건너뛰면서 부호화하는 대신에, 지정 시간 간격 내에서 가변 간격으로 화면을 부호화 하는 것을 지원한다.

그림 1은 화면의 가능한 부호화 모드와 목표 비트율을 나타낸다.

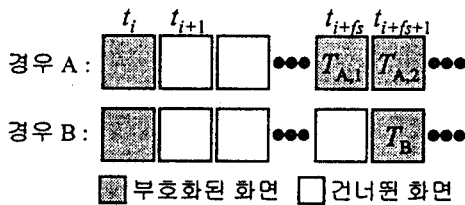


그림 1. 부호화 가능 모드

그림 1에서  $f_s$ 는 마지막으로 부호화된 화면과 다음으로 부호화 되어야 할 화면 사이의 간격을 나타낸다. 그림 1에서  $t = t_{i+f_s+1}$ 까지의 화면은 현재 화면에 할당된 목표 비트량이 이전 부호화 화면의 헤더 정보를 부호화하는데 사용된 비트량보다 부족하여 강제적으로 건너뛰어지게 되는 화면이며, 마지막 두 화면은 부호화 화면의 특성에 의해 모드가 결정되는 화면이다.

경우 A와 경우 B에 대해 부호화 왜곡은 각각 다음과 같이 정의된다.

$$\bar{D}_{A,[t_{i+1}, t_{i+f_s}]} = \sum_{j=i+1}^{i+f_s-1} D_s(Q_i, j) + D_c(Q_{i+f_s}) + D_c(Q_{i+f_s+1}) \quad (4)$$

$$\bar{D}_{B,[t_{i+1}, t_{i+f_s}]} =$$

$$\sum_{j=i+1}^{i+f_s-1} D_s(Q_i, j) + D_s(Q_i, i+f_s) + D_c(Q_{i+f_s+1}) \quad (5)$$

경우 A에서는  $t = t_{i+f_s+1}$ 의 화면이 부호화된다고 가정하여 전체 왜곡을 계산하였지만, 이 화면에 할당되는 목표 비트량이 이전 부호화 화면의 헤더 정보, 움직임 정보, 모양 정보를 표현한 비트량보다 적은 경우에는 비트율 제어 관점에서 건너뛰어져야 한다. 따라서 이 경우에는 식 (4)의 두 번째 부호화 화면 왜곡을 계산하는  $D_c(Q_{i+f_s+1})$ 를  $D_s(Q_{i+f_s}, i+f_s+1)$ 으로 변경하여 경우 A의 전체 부호화 왜곡을 계산한다.

본 논문에서는 객체기반 부호화에 대해 부호화 왜곡을 다음과 같이 정의한다.

$$d_A(Q, f_s) = \sum_{vo\_id=1}^M \bar{D}_{A,[t_{i+1}, t_{i+f_s}]}(Q, f_s) \quad (6)$$

$$d_B(Q, f_s) = \sum_{vo\_id=1}^M \bar{D}_{B,[t_{i+1}, t_{i+f_s}]}(Q, f_s) \quad (7)$$

여기서  $M$ 은 부호화 객체의 수를 나타낸다.

만약 화면과 화면 사이의 화질의 변화가 급격하게 변하는 것을 허용한다면, 위에서 제안한 부호화 왜곡 모델과 비트율 제어 알고리즘을 이용하여 현재 비디오 객체들의 양자화 변수를 정하면 된다. 하지만 화면과 화면사이의 급격한 화질의 변화는 사람의 시각에 민감하게 반응하므로, 제안된 왜곡 모델에 화질의 급격한 열화를 방지할 수 있는 방법이 필요하다. MPEG-4 VM5 비트율 알고리즘 [4]의 경우, 현재 화면의 양자화 변수  $Q$ 는 이전 부호화 화면의 양자화 변수  $Q'$ 에 대해 다음과 같은 범위 내에서 결정된다.

$$Q' - \frac{1}{4}Q' \leq Q \leq Q' + \frac{1}{4}Q' \quad (8)$$

제안된 알고리즘에서도 현재 부호화 객체의 양자화 변수는 식 (8)의 제한 범위 내에서 정해지며, 양자화 변수의 제한 때문에 경우 A와 비교하여 항상 경우 B에서 2배의 성능을 얻지 못한다. 즉, 경우 A에서 첫번째 부호화 화면에 할당된 비트수  $T_{A,1}$ 과 두 번째 부호화 화면에 할당되는 비트수  $T_{A,2}$ 의 합은 경우 B에서 부호화 화면에 할당되는 비트수  $T_B$ 와 항상 같지 않다

### 4. 비트율 제어 알고리즘

본 논문에서 제안한 왜곡 모델을 이용하는 부호화 성능을 최적화하면서 버퍼 넘침이나 결핍 현상을 방지하기 위해서 다음과 같은 조건이 만족되어야 한다.

$$\bar{R} \leq R \quad (9)$$

$$R_{A,1} \equiv \begin{cases} B_i + R(t_{i+f_s}) < B_{max} \\ B_i + R(t_{i+f_s}) - f_s \cdot R_{drain} > 0 \end{cases} \quad (10)$$

$$R_{A,2} \equiv \begin{cases} B_{i+f_s} + R(t_{i+f_s+1}) < B_{max} \\ B_{i+f_s} + R(t_{i+f_s+1}) - (f_s + 1) \cdot R_{drain} > 0 \end{cases} \quad (11)$$

$$R_{B,1} \equiv \begin{cases} B_i < B_{max} \\ B_i - f_s \cdot R_{drain} > 0 \end{cases} \quad (12)$$

$$R_{B,2} \equiv \begin{cases} B_i + R(t_{i+f_s+1}) < B_{max} \\ B_i + R(t_{i+f_s+1}) - (f_s + 1) \cdot R_{drain} > 0 \end{cases} \quad (13)$$

여기서  $R$ 은 목표 비트율,  $B_{max}$ 는 버퍼의 크기,  $B_i$ 는 현재 버퍼의 상태,  $R_{drain}$ 은 VOP당 버퍼를 빠져나가는 양을 나타낸다. 위의 조건을 만족시키기 위해 다음 알고리즘을 사용한다.

- ①  $f_s$ 의 값을 1로 설정하고, 현재 남아있는 비트수  $T_n$  이전 VOP를 부호화하는데 사용된 비트수  $T_p$ , 남아있는 VOP의 개수  $N$ 을 이용하여 현재 VOP에 할당 가능한 비트수를 계산한다.
- ② 할당된 비트수를 현재 버퍼 상태와 전체 버퍼 크기를 고려하여 버퍼 넘침이나 결핍이 발생하지 않도록 조절한다. 조절된 비트수가 이전 VOP의 헤더 정보, 움직임 정보, 모양 정보를 부호화하는데 사용된 비트수보다는 적은 경우에는 현재 할당된 비트수로는 현재 VOP를 부호화하는데 적절하지 않으므로  $f_s$ 의 값을 증가시키면서 다시 조절한다.
- ③ 각 객체의 부호화 성능을 최대로 하기 위해, VOP에 할당된 비트량을 각 객체의 특성에 알맞게 분배하고 각 객체에 할당된 비트량과 이전 해당 비디오 객체의 양자화 변수(QP)를 고려하여 현재 비디오 객체의 QP를 결정한다.
- ④ Case I과 Case II에 대한 부호화 왜곡을 계산하고, 식 (9)의 버퍼 조건을 만족하는지 검사한다. 버퍼 조건을 만족할 때까지  $f_s$ 값을 증가시키면서 현재까지의 과정을 반복한다.
- ⑤  $d_i(Q, f_s)$ 가 더 작은 부호화 왜곡을 가진다면, 각 VO의 현재 QP값을 이용하여 VO들을 부호화하고,  $d_{if}(Q, f_s)$ 이 작은 경우에는 현재 VOP는 건너뛰므로써 얻은 추가 할당 비트를 다음 VOP에 더하기 위해 부호화해야 할 VOP의 수를  $N_s = (N_s - f_s + 1)/2$ 으로 갱신하고, 이 값을 이용하여 타당한 양자화 변수를 정한 후에 해당 VOP를 부호화한다.

식 (9)에서 식 (13)까지의 버퍼 조건과 각 부호화 화면의 버퍼 상태를 이용한 비트율 제어를 위

해 그림 2에 제안된 알고리즘을 이용한다.

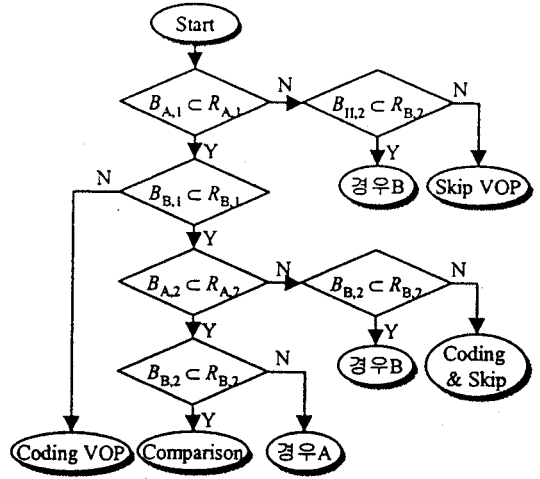


그림 2. 비트율 제어를 위한 버퍼 체크

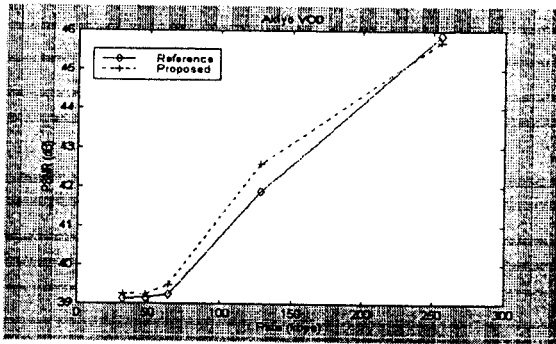
그림 2에서  $B_{A,1}$ ,  $B_{A,2}$ ,  $B_{B,1}$ ,  $B_{B,2}$ 는  $t = t_{i+f_s}$ 와  $t = t_{i+f_s+1}$ 에서 각각 경우 A와 경우 B에 해당하는 화면의 버퍼 상태를 나타낸다.

### 5. 실험 결과 및 분석

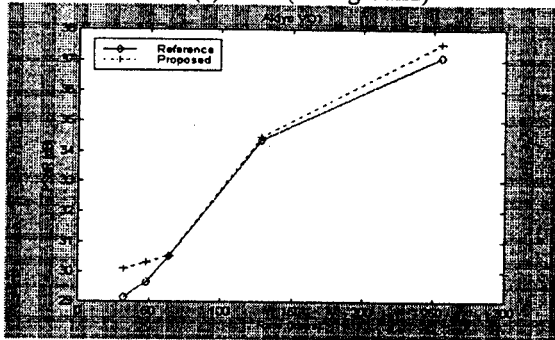
본 논문에서 제안된 알고리즘의 성능을 검사하기 위해 본 논문은 2개의 비디오 객체를 포함하면서 300 화면으로 구성된 CIF 형식의 AKIYO 시퀀스를 이용하였다. 부호화 성능의 비교를 위해 MPEG-4 참조 알고리즘(VM5) [5]과 제안된 알고리즘을 이용하여 32 kbps에서 256 kbps의 비트율로 부호화한 결과를 서로 비교하였다.

그림 3은 비트율-왜곡 곡선을 나타낸 것이다. 그림 3에서 볼 수 있듯이, 제안된 알고리즘이 VM5보다 성능이 우수함을 알 수 있다. 특히, 낮은 비트율에서 VM5는 단지 버퍼의 상태에 의해서만 화면이 건너뛰어지지만, 제안된 알고리즘에서는 버퍼 상태 뿐만 아니라 최소 왜곡 모델을 이용하기 때문에 더 나은 성능을 제공한다.

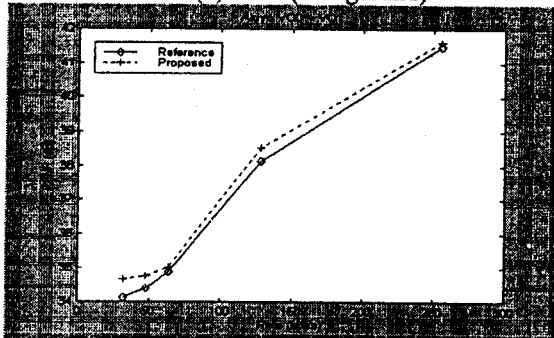
실험을 통해, 제안된 알고리즘은 VM5와는 달리 실제 발생된 비트량이 목표 비트량과 거의 일치함을 보였다. 그림 4는 각각의 비트율에서 버퍼의 상태를 나타낸 것이다. 그림 4에서 알 수 있듯이, 제안된 알고리즘에서는 낮은 비트율과 높은 비트율에서 모두 안정적임을 알 수 있다. 이 실험을 통해 제안된 알고리즘에서 버퍼는 평균적으로 50% 정도의 상태를 유지하면서 ±20% 정도의 변동을 보였다.



(a) VO0 (Background)

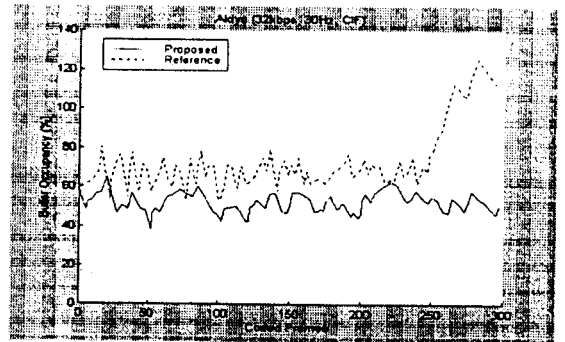


(b) VO1 (Foreground)

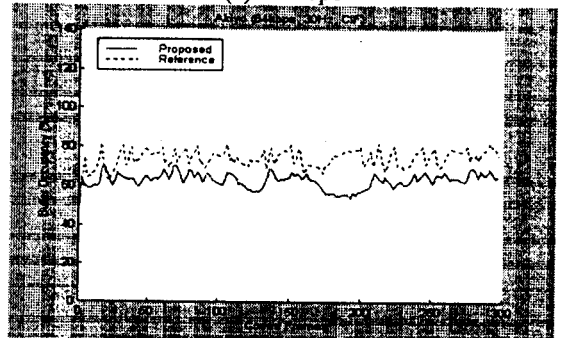


(c) VO0 + VO1

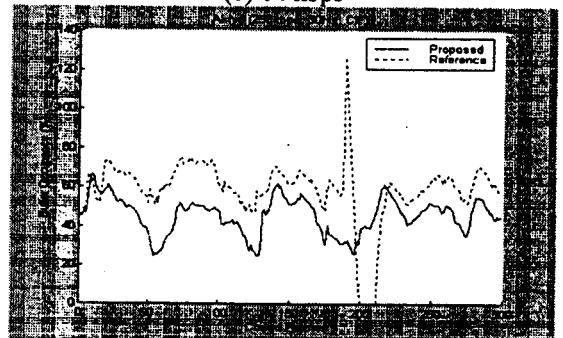
그림 3. R-D 곡선 비교



(a) 32 kbps



(b) 64 kbps



(c) 256 kbps

그림 4. 버퍼 상태 비교

## 6. 결론

본 논문에서는 화면기반 왜곡 모델을 객체기반 왜곡 모델로 확장하고, 부호화 성능을 높일 수 있는 비트율-왜곡 모델과 버퍼 조건을 제안하였다. 또한 새로운 부호화 모드를 결정하기 위한 부호화 알고리즘과 비트율 제어 알고리즘을 제안하였다. 본 논문에서 제안된 알고리즘은 목표 비트율에 거의 비슷하게 부호화 비트를 발생시키면서 기존의 방법에 비해 더 나은 성능을 제공하며 안정된 버퍼 상태를 유지한다.

## 감사의 글

본 연구는 광주과학기술원(K-JIST) 초고속광네트워크연구센터(UFON)를 통한 한국과학재단 우수연구센터(ERC)와 교육부 두뇌한국21 (BK21) 정보기술사업단의 지원에 의한 것입니다.

## 참고문헌

- [1] ISO/IEC 14496-2, "Information technology - coding of audio/visual objects," Oct. 1998.
- [2] A. Vetro, Y. Wang and H. Sun, "Estimating distortion of coded and non-coded frames for frameskip-optimized video coding," *Int'l Conf. on Multimedia and Expo*, Jan. 2001.
- [3] A. Vetro, Y. Wang and H. Sun, "Rate-Distortion Optimized Video Coding Considering Frame-skip," *ICIP '01*, pp. 534-537, Oct. 2001.
- [4] A. Vetro, H. Sun, and Y. Wang, "MPEG-4 rate control for multiple video objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 186-199, Feb. 1999.
- [5] ISO/IEC 14496-5, "Information technology - coding of audio/visual objects," July 2000.