

시공간 왜곡을 이용한 MPEG-4 객체기반 비트율 제어 방법

이 정 우, 호 요 성^o

광주과학기술원 정보통신공학과

MPEG-4 Object-based Rate Control Using Spatio-Temporal Distortion

Jeong-Woo Lee and Yo-Sung Ho^o

Kwangju Institute of Science and Technology (K-JIST)

{jeongwoo, hoyo}@kjist.ac.kr

요 약

본 논문은 MPEG-4 표준을 따르는 객체기반 비디오 부호화에서 고정 비트율을 지원하는 알고리즘을 기술한다. 각각의 비디오 객체가 서로 다른 프레임율을 가지면서 독립적으로 부호화될 수 있는 경우에 비트율과 버퍼에 대한 제한 조건을 살펴보고, 부호화되는 객체의 왜곡과 건너뛰어지는 객체의 왜곡을 계산하여 현재 부호화 시간에서 부호화되어야 하는 객체의 최적 집합(optimal set)을 결정하는 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. 서 론

H.263, MPEG-1, MPEG-2, MPEG-4와 같은 비디오 부호화 표준들은 가변길이 부호화(Variable Length Coding) 방법을 사용하기 때문에 비디오 부호화 과정을 통해 각 화면마다 발생하는 비트수가 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 크기 조절을 통해 비트 발생량을 제어하는 MPEG-1이나 MPEG-2 표준과는 달리, H.263과 MPEG-4 표준에서는 비트 발생량을 제어할 수 있는 방법으로 화면의 건너뛰음을 지원한다. 부호화 과정에서 건너뛰어진 화면은 복호화 과정에서 이전에 복원된 화면을 그대로 다시 보여주기 때문에, 공간적 화질 이외에 시간적 화질도 부호화 성능을 측정하는 중요한 요소가 된다.

현재까지 제안된 알고리즘들은 화면의 건너뛰음이 단지 버퍼 상태에 따라 결정되기 때문에 복원 후에 재구성된 건너뛰어진 화면들은 화질의 열화를 초래하게 된다. 따라서 버퍼 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 수행하므로 MPEG-4 비트율 제어 알고리즘에서는 모양 정보를 부호화해서 발생하는 비트량도 중요하

게 고려해야 한다. 또한 MPEG-4 비디오 부호화 알고리즘이 지원하는 독립적인 비디오 객체 부호화를 지원하는 알고리즘의 개발이 필요하다.

최근에 A. Vetro에 의해 부호화 화면과 건너뛰어진 화면의 왜곡을 예측할 수 있는 방법과 비트율 제어 알고리즘이 제안되었다 [2,3]. 이 방법은 공간적 화질과 시간적 화질 사이의 절충(trade-off)을 통해 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 건너뛰음 비율(frame skip rate)을 계산하기 위해 정해진 최대값 내에 있는 모든 값에 대해 왜곡을 계산하기 때문에 많은 연산이 요구된다.

공간적 화질과 시간적 화질 사이의 절충을 고려하면서, 다중 객체 부호화를 지원하는 알고리즘 [4, 5]은 적은 연산량으로 기존의 방법보다 우수한 성능을 제공하지만, 화면의 건너뛰음이 화면 단위로 수행되기 때문에 최적의 성능을 제공할 수 없는 단점이 있다.

본 논문에서는 각각의 객체를 독립적으로 부호화하면서 주어진 목표 비트율에 알맞게 비트량을 발생시키고, 버퍼 제한 조건을 만족하면서 기존의 방법보다 나은 성능을 제공할 수 있는 알고리즘을 제안한다. 제안된 방법을 이용하여 최적의 R-D 성능을 각각의 객체가 유지하도록 부호화된다.

2. 가변 프레임율 객체 부호화 모델

가변 프레임율 객체 부호화 모델에서는 각각의 객체에 대해 다른 프레임 건너뛰임율과 상응하는 객체의 양자화 변수를 자유롭게 선택할 수 있지만, 부호화되는 객체들의 시간 정보를 모두 추적해야 하기 때문에 비트율 제어 방법이 복잡하게 된다.

그림 1은 부호화 비디오 객체의 수가 3개인 경우에 부호화될 수 있는 형태를 보여준다.

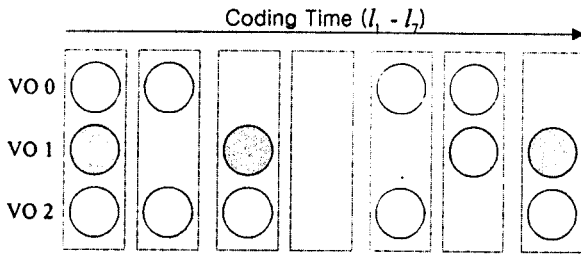


그림 1. 객체단위 부호화 예

그림 1에서 볼 수 있듯이, 매 부호화 시간에서 모든 객체가 부호화되거나 건너뛰어질 필요가 없다. 그림 1과 같이 부호화되기 위해서는 고려하는 모든 부호화 시간내에서 매 부호화 시간마다 발생량과 버퍼에 대한 제한 사항을 모두 만족해야 한다. 본 절에서는 부호화 성능을 고려하기 전에 비트율과 버퍼 관점에서 제한 사항을 살펴본다.

2.1 발생량 제한 사항

M을 부호화하고자 하는 모든 객체들의 집합이라고 하고, L을 고려하는 시간내에 있는 모든 부호화 시간들의 집합이라고 하자. 그리고 M_l은 L에 속하는 모든 l에 대해 t = t_l에서 부호화되는 객체들의 집합이라고 하자. 이 경우에 그림 1에서 부호화된 객체들에 대한 M, L, M_l은 그림 2와 같이 표현된다.

주어진 시간내에서 발생하는 비트량은 다음의 제한 사항을 만족해야 한다.

$$\sum_{l \in L} \sum_{j \in M_l} R_j(t = t_l) \leq R_{budget} \tag{1}$$

여기서 R_j(t = t_l)은 시간 t = t_l에서 j번째 비디오 객체를 부호화한 후에 발생한 비트량이고, R_{budget}은 전체 목표 비트량을 기준으로 주어진 시간내에서 부호화 가능한 최대 비트량이다.

- M = {0, 1, 2}
- L = {l₁, l₂, l₃, l₄, l₅, l₆, l₇}
- M_l(l = l₁) = {0, 1, 2}
- M_l(l = l₂) = {0, 2}
- M_l(l = l₃) = {1, 2}
- ...

그림 2. M, L, M_l의 표현

2.2 버퍼 제한 사항

모든 부호화 시간 간격에 대해 버퍼 넘침 현상과 결핍 현상을 피하기 위해, 매 순간 발생하는 비트량은 다음과 같은 조건을 만족해야 한다.

$$B_{i+l_0} + \sum_{j \in M_l} R_j(t = t_{l_0} : t_l) < B_{max}; \forall l \in L \tag{2}$$

$$B_{i+l_0} + \sum_{j \in M_l} R_j(t = t_{l_0} : t_l) - (l - l_0)R_{drain} > 0; \forall l \in L \tag{3}$$

여기서 i는 현재 기준 시점, B_{i+l₀}는 부호화를 시작하기 전의 현재 버퍼 상태를 나타내고, R_{drain}은 매 순간마다 버퍼를 통해 전송로를 통해 빠져나가는 비트율이고, R_j(t = t_{l₀} : t_l)는 현재부터 시간 t = t_l까지 j번째 비디오 객체를 부호화한 후에 발생한 비트량이다. B_{max}는 최대 버퍼 크기이다.

2.3 구현상의 제한 사항

앞에서 각각의 객체가 가변 프레임율을 가질 수 있도록 비트량과 버퍼에 관한 제한사항을 제시하였지만, 작은 크기의 L에 대해서도 실제로 모든 경우의 수를 고려하는 것은 구현상의 제한이 따른다. 현재 시점에서 미래에 부호화되는 객체에서 발생하는 비트량을 정확하게 예측하는 것도 불가능하며, 할당 가능한 모든 목표 비트량에 대해 최적의 객체 집합을 결정한다는 것은 대단히 어려운 문제이다.

물론 그림 1의 경우, t = t_l에서 t = t_l까지 모든 부호화 시간에 대해 할당 가능한 목표 비트량을 가지고 실제 부호화 과정을 수행한 후에 최대의 성능을 제공하는 객체들의 집합을 선택하는 것도 가능하지만, 매 시간마다 상당히 많은 부호화 지연이 발생할 뿐만 아니라 부호화하는데 너무 많은 시간이 요구된다.

각 객체가 가변 프레임율을 가지도록 지원하는 간단한 방법은 현재 부호화 시간에서 버퍼 넘침이나 결핍 현상을 방지하기 위해, 현재 버퍼의 상태를 이용하여 할당 가능한 목표 비트율을 결정하고, 할당된 목표 비트량을 가지고 최대의 비트율-왜곡(R-D) 곡선을 지원하는 비디오 객체의 집합을 결정하는 것이다.

3. 가변 프레임율 제한 객체 부호화 모델

그림 3는 본 논문에서 고려하는 가변 프레임율 제한 객체 부호화 모델을 보여준다. 그림 3에서 볼 수 있듯이, 제안된 알고리즘은 현재 부호화 시간에서 부호화해야 할 객체들과 건너뛰어야 할 객체들을 결정하고, 현재 할당된 목표 비트량을 기준으로 각 부호화 객체의 양자화 변수를 정하는 방법이다.

이를 구현하기 위해 각각의 부호화 객체에 목표 비트량을 할당하는 방법과 효과적인 성능을 가질 수 있도록 최적의 부호화 객체 집합을 선택하는 방법이 제안되어야 한다.

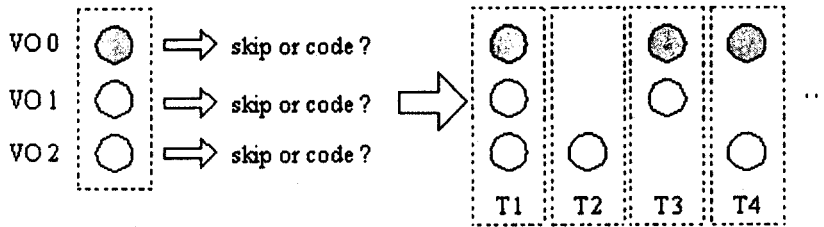


그림 3. 가변 프레임율 제한 객체 모델

3.1 버퍼 모델

MPEG-4 비트율 제어 알고리즘에서는 모양 정보를 부호화해서 발생하는 비트량도 중요하게 고려해야 하기 때문에, 모양 정보를 고려한 비트율 제어 알고리즘에 의해 강제로 건너뛰어지는 VOP의 수를 고려해서 버퍼 제어를 해야 한다. 이 경우에 발생 비트량은 다음의 조건을 만족해야 한다.

$$B_{i+l_0} + \sum_{j \in M(i+f_s)} R_j(t_{i+f_s}) < B_{max} \quad (4)$$

$$B_{i+l_0} + \sum_{j \in M(i+f_s)} R_j(t_{i+f_s}) - (f_s - l_0)R_{drain} > 0 \quad (5)$$

여기서 f_s 는 모양 정보 비트량을 고려하여 부호화되어야 하는 시간 정보를 나타내고, $M(i+f_s)$ 는 시간 $t=i+f_s$ 에서 부호화되는 객체들의 집합이다.

3.2 비트율-왜곡 모델

현재 부호화 시간에서 최적의 비디오 객체 집합을 선택하기 위해서는 각 객체의 부호화 효율을 계산하는 것과 현재 VOP의 부호화 왜곡을 계산하는 것이 필요하다 [2-5].

영상 부호화 방식에서 부호화되는 j 번째 객체의 왜곡은 다음과 같이 표현된다.

$$D_{j,c}(Q_i) = a \cdot 2^{-2R_j(t_i)} \cdot \sigma_{z_{i,j}}^2 \quad (6)$$

여기서 $\sigma_{z_{i,j}}^2$ 는 객체 분산값, $R_j(t_i)$ 는 j 번째 객체 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, j 번째 객체의 k 번째 건너뛰어진 화면의 왜곡은 다음과 같이 표현된다.

$$D_{j,s}(Q_i, k) = D_{j,c}(Q_i) + E_j \{ \Delta^2 z_{i,k} \} \quad (7)$$

여기서 $E_j \{ \Delta^2 z_{i,k} \}$ 는 j 번째 객체의 i 번째 화면과 k 번째 화면 사이의 예측 오류이며, optical flow를 이용하여 다음과 같이 정의된다.

$$E_j \{ \Delta^2 z_{i,k} \} = \sigma_x^2 \sigma_{\Delta x_{i,k}}^2 + \sigma_y^2 \sigma_{\Delta y_{i,k}}^2 \quad (8)$$

여기서 σ_x^2 와 σ_y^2 는 각각 j 번째 객체의 i 번째 화면에서 x 와 y 방향으로 기울기 값들의 분산 값을 나타내고, $\sigma_{\Delta x_{i,k}}^2$ 와 $\sigma_{\Delta y_{i,k}}^2$ 는 x 와 y 방향으로 움직임 벡터들의 분산 값을 나타낸다.

부호화되는 객체들의 집합 $M(i+f_s)$ 에 대해, 현재

VOP의 부호화 왜곡은 다음과 같이 정의된다.

$$d_{M(i+f_s)}(Q, f_s) = \sum_{j \in M(i+f_s)} D_{j,c}(Q_{i+f_s}) + \sum_{j \in M(f_s)} D_{j,s}(Q_i, f_s) \quad (9)$$

가장 좋은 R-D 곡선을 제공하는 $M(i+f_s)$ 집합을 결정하기 위해 그림 2에 있는 M 과 같이, 부호화 객체를 구성하는 집합 M 의 모든 부분집합에 대해 식 (9)의 부호화 왜곡을 계산하여 가장 적은 왜곡을 가지는 것으로 결정한다.

$$\min_{M(i+f_s) \subset M} |d_{M(i+f_s)}(Q, f_s)| \quad (10)$$

4. 비트율 제어 알고리즘

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

$$\min_{M(i+f_s) \subset M} |d_{M(i+f_s)}(Q, f_s)| \quad (11)$$

$$\text{충족버퍼조건} \begin{cases} \bar{R} \leq R \\ B_i + \sum_{j \in M(i+f_s)} R_j(t_{i+f_s}) < B_{max} \\ B_i + \sum_{j \in M(i+f_s)} R_j(t_{i+f_s}) - f_s \cdot R_{drain} > 0 \end{cases} \quad (12)$$

여기서 R 은 목표 비트율, B_{max} 는 버퍼의 크기, B_i 는 현재 버퍼의 상태, R_{drain} 은 VOP당 버퍼를 빠져 나가는 양을 나타낸다.

비트율 제어와 최적의 부호화 객체 집합을 구하기 위해 그림 4의 알고리즘을 사용한다. 화면기반 다중객체 부호화 알고리즘의 경우, 현재 할당된 목표 비트량이 이전 부호화 화면의 헤더 정보, 모양 정보, 움직임 정보를 표현한 비트량보다 적으면, 강제로 현재 VOP를 건너뛰게 한다. 그러나 제안된 알고리즘의 경우, 객체 단위로 독립적으로 부호화가 가능하기 때문에 전체 객체의 일부분만 부호화가 가능하다.

제안된 알고리즘에서 목표 비트량은 현재 시간에서 모든 비디오 객체들이 부호화된다고 가정하여 다음과 같이 결정된다.

$$T_B = T_V \cdot \frac{\bar{B}_i + 2(B_{max} - \bar{B}_i)}{2\bar{B}_i + (B_{max} - \bar{B}_i)} \quad (13)$$

여기서 T_V 는 그림 4의 두 번째 과정에 의해 결정된 값이고, T_B 는 세 번째 과정에 의해 결정된 값이다.

- ① 입력으로 들어오는 모든 비디오 객체를 부호화하는 것을 기본으로 하기 위해, 초당 30 화면을 기준으로 f_s 의 값을 1로 설정한다. 초당 15 화면을 부호화하는 경우에 f_s 의 값은 2로 결정되고 증가하는 단위는 2이다.
- ② 현재 남아있는 비트량, 이전 VOP를 부호화하는데 사용된 비트량, 남아있는 VOP의 개수를 이용하여 현재 VOP에 할당 가능한 비트량(T_v)을 계산한다.
- ③ 할당된 비트량을 현재 버퍼 상태와 전체 버퍼 크기를 고려하여 버퍼 넘침이나 결핍이 발생하지 않도록 조절한다(T_b).
- ④ 할당된 최종 비트량과 이전 VOP의 헤더 정보, 움직임 정보, 모양 정보를 부호화하는데 사용된 비트량을 비교하여 부호화 가능한 객체 부호화 집합 M_L 을 결정한다. 따라서 목표 비트량의 부족으로 화면을 건너뛰어야 하는 경우에 부호화 가능 객체 집합 M_L 은 공집합(\emptyset)으로 정의된다. 이 경우에 f_s 를 증가시키고 다시 목표 비트량을 계산한다.
- ⑤ M_L 에 속하는 $M(f_s)$ 에 대해, 각 객체의 부호화 성능을 최대도 하기 위해, 현재 화면에 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.
- ⑥ 식 (9)의 모델을 이용하여 현재 $M(f_s)$ 의 부호화 왜곡을 계산하고, 식 (12)의 버퍼 조건을 만족하는지 검사한다.
- ⑦ M_L 에 속하는 모든 $M(f_s)$ 에 대해 위의 과정을 반복한 후에, 버퍼 조건을 만족하면서 최소의 부호화 왜곡을 가지는 부호화 객체 집합 $M(f_s)$ 을 결정한다.
- ⑧ 정해진 f_s 를 이용하여 다음에 부호화해야 하는 부호화 시간(coding time instant)을 계산하고, $M(f_s)$ 에 속하는 비디오 객체에 대해서만 해당 객체의 양자화 변수(Q_{i+f_s})를 이용하여 부호화 과정을 수행한다.

그림 4. 비트율 제어 알고리즘

각 객체의 부호화 성능을 최대도 하기 위해, 현재 화면에 할당된 비트량을 각 객체의 분산값과 움직임 정보의 비율과 해당 객체의 이전 부호화 화면에서 모양 정보를 부호화하여 발생한 비트량을 이용하여 각 객체에 할당한다 [5].

각 객체에 할당된 비트량을 이용하여 결정되는 양자화 변수는 화면과 화면사이의 급격한 화질 변화가 발생하지 않도록 이전 부호화 화면의 양자화

변수값과 관련한다. 따라서 현재 버퍼 상태에 따라 결정된 목표 비트량 계산 과정과는 별도로 버퍼 넘침이나 결핍 현상이 발생하지 않는지 검사하는 과정이 필요하다.

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) [7]과 제안된 알고리즘을 이용하여 32 kbps에서 256 kbps의 비트율로 부호화한 결과를 서로 비교하였다.

제안된 알고리즘의 성능을 평가하기 위해, 부호화 과정에서 목표로 하는 비트율에 정확하게 실제 비트량이 발생하는지를 검사하고, 비트율 제어 방법에서 우선적으로 요구되는 버퍼 넘침 현상과 결핍 현상이 일어나는지 알아보기 위해 버퍼 상태를 측정한다. 버퍼를 정확하게 제어하면서 기존의 방법에 비해 우수함을 보이기 위해 기존의 방법과 제안된 알고리즘의 R-D 곡선을 비교한다.

그림 5은 32kbps에서 256 kbps까지 설정된 목표 비트율로 VM5 알고리즘과 제안된 알고리즘으로 부호화한 후에 목표 비트율과 실제 부호화 비트율 사이의 관계를 나타낸 것이다. 그림 5에 보인 것처럼, 기존의 알고리즘은 모든 비트율에서 목표 비트율보다 많은 비트량을 발생시키는 것에 반해, 제안된 알고리즘은 목표 비트율에 거의 비슷하게 발생하는 것을 확인하였다.

그림 6은 각각의 비트율에서 전체 300 화면에 대해 화면당 버퍼 상태를 나타낸 것이다. 그림 7에서 알 수 있듯이, 기존의 알고리즘은 32 kbps와 256 kbps에서 버퍼 넘침 현상이 발생하고, 256 kbps에서는 버퍼 결핍 현상도 발생하는 것을 확인할 수 있지만, 제안된 알고리즘에서는 비트율에 상관없이 모든 비트율에서 모두 안정적임을 알 수 있다. 이 실험을 통해 제안된 알고리즘에서 버퍼는 평균적으로 60% 정도의 상태를 유지하면서 $\pm 20\%$ 정도의 변동 을 보였다.

그림 7은 기존 알고리즘과 제안된 알고리즘의 비트율-왜곡 곡선을 나타낸 것이다. 그림 8에서 볼 수 있듯이, 제안된 알고리즘이 VM5보다 성능이 우수함을 알 수 있다. 특히, 모든 비트율에서 VM5는 단지 버퍼의 상태에 의해서만 화면이 건너뛰어지지 만, 제안된 알고리즘에서는 버퍼 상태 뿐만 아니라 최소 왜곡 모델을 이용하기 때문에 더 나은 성능을 제공한다.

배경과 같은 객체에는 적은 비트량을 할당하고, 움직임이 많은 객체에는 많은 비트량을 할당함과 동시에 움직임이 많은 객체의 화면을 더 작은 프레임율로 부호화함으로써 기존의 방법에 비해 거의 1 ~ 2dB 정도의 성능 향상을 가져온다.

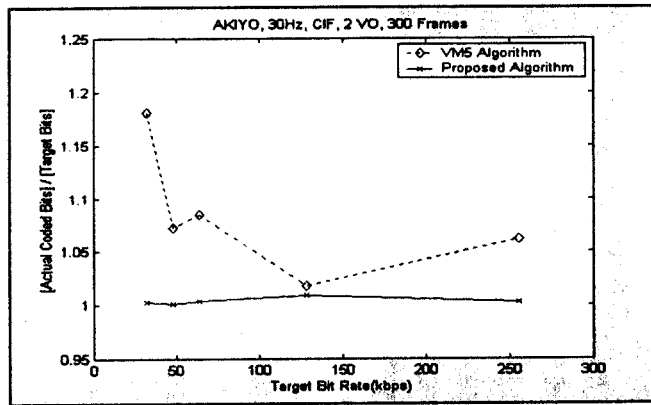
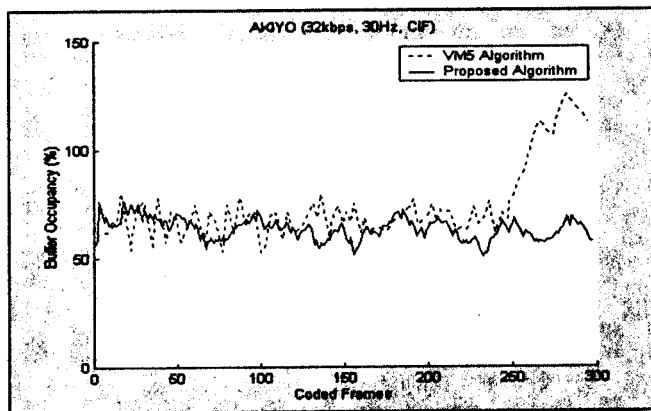
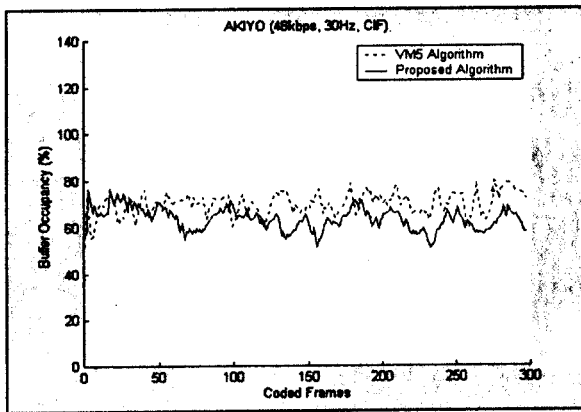


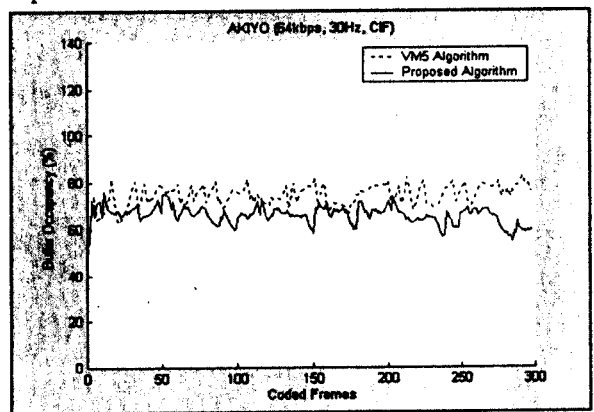
그림 5. 목표 비트량과 실제 비트 발생량 비교



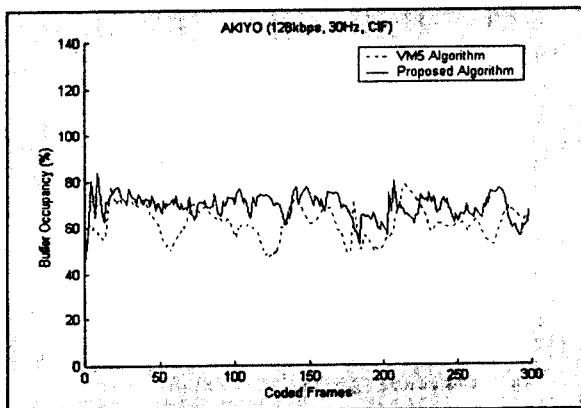
(a) 32 kbps



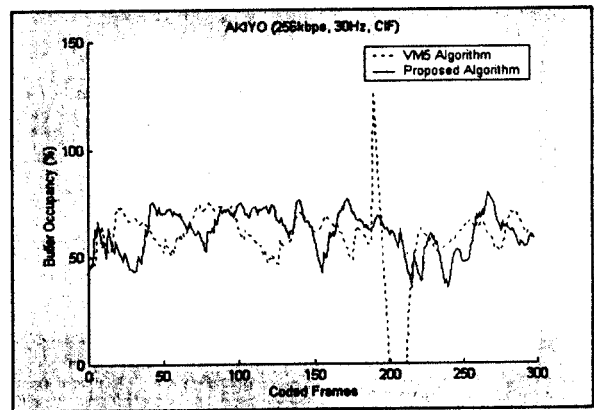
(b) 48 kbps



(c) 64 kbps

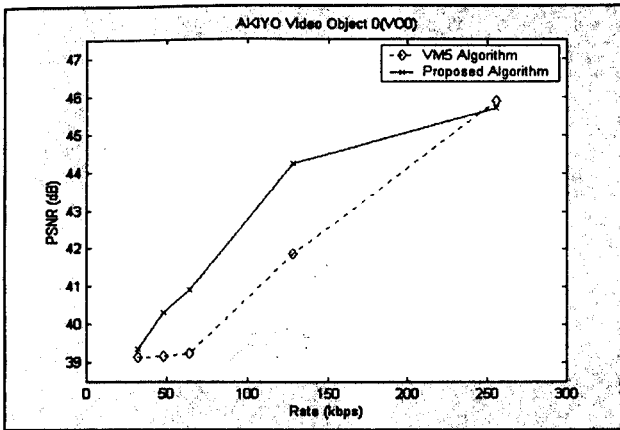


(d) 128 kbps

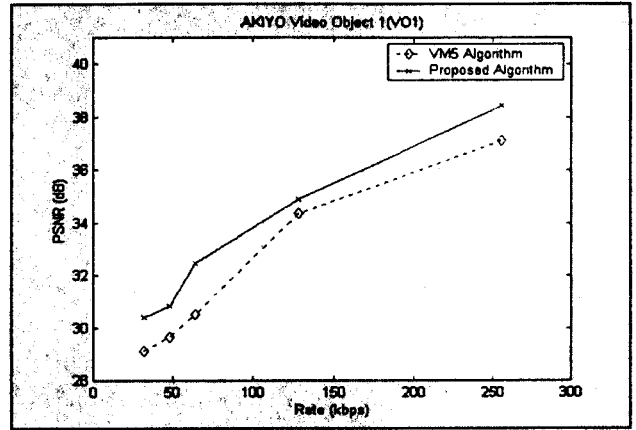


(e) 256 kbps

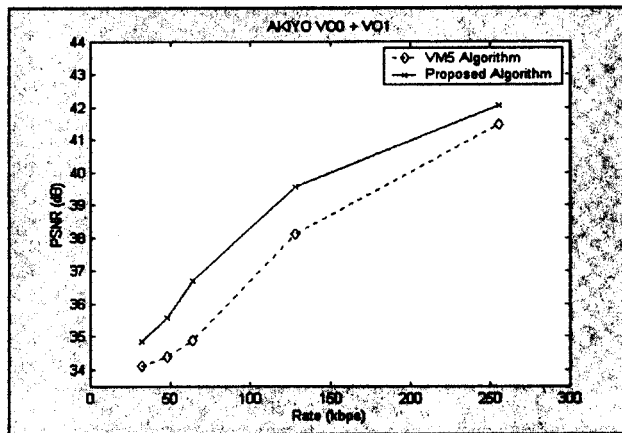
그림 6. 버퍼 상태 비교



(a) VO 0



(b) VO 1



(c) VO 0 + VO 1

그림 7. R-D 곡선 비교

6. 결론

본 논문에서는 각각의 비디오 객체가 가변 프레임율을 가지는 경우에 비트율과 버퍼에 대한 조건을 소개하고, 부호화 복잡도를 줄이면서 가변 프레임율 기반 객체 부호화를 위한 비트율 제어 방법을 제안하였다. 기존의 방법에 비해 향상된 R-D 곡선을 제공하기 위해, 매 부호화 시간에서 각각의 객체는 제안된 비트율-왜곡 모델에 의해 부호화되는지 건너뛰어지는지 결정된다. 제안된 알고리즘은 목표 비트율에 거의 비슷하게 부호화 비트를 발생시키면서, 기존의 방법에 비해 더 나은 성능을 제공하며 안정된 버퍼 상태를 유지한다. 각각의 객체가 가변 프레임율을 가지기 때문에 부호화 과정에서 합성 문제(composition problem)이 발생할 수 있으므로 이에 대한 해결책이 요구된다.

감사의 글

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

7. 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] 이정우, 호요성, "MPEG-4 비디오 부호화를 위한 객체기반 비트율 제어 방법," 컴퓨터사이언스추계학술발표 논문집, pp. 143-146, 2001.
- [5] J.W. Lee, A. Vetro, Y. Wang and Y. S. Ho, "Object-based Rate Allocation with Spatio-Temporal Trade-offs", *VCIP 2002*, Jan. 2002.
- [6] 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.
- [7] ISO/IEC 14496-5, "Information technology - coding of audio/visual objects," July 2000.