

논문 2004-41SP-1-6

시공간 화질의 절충을 고려한 MPEG-4 비트율 제어 알고리즘

(MPEG-4 Rate Control Method with Spatio-Temporal Trade-Offs)

이 정 우*, 호 요 성**

(Jeong-Woo Lee and Yo-Sung Ho)

요 약

본 논문에서는 여러 개의 비디오 객체를 부호화할 때 기존의 MPEG-4 VM5 알고리즘에 비해 부호화 성능을 향상시키면서 고정 비트율을 지원하는 새로운 알고리즘을 기술한다. 특히, 부호화되는 객체와 건너뛰어지는 객체의 왜곡을 예측하는 비트율-왜곡 모델을 제시하고, 부호화 성능을 향상시키기 위해 시간적-공간적 절충을 고려한 부호화 모드와 버퍼를 제어하는 방법을 제안한다. 본 논문에서 제안된 객체 기반의 비트율 제어 방법은 넓은 범위의 비트율에서 향상된 성능을 보이며, 목표 비트율에 근접한 압축 비트열을 발생하고, 버퍼 상태를 안정적으로 조절하기 때문에 버퍼 넘침과 버퍼 결핍 현상이 발생하지 않는다.

Abstract

This paper describes a new bit allocation algorithm that can achieve a constant bit rate when coding multiple video objects, while improving rate-distortion (R-D) performance over the VM5 method for MPEG-4 object-based video coding. In particular, we propose two models to estimate the rate-distortion characteristics of coded objects as well as skipped objects. Based on the proposed models, we present several R-D coding modes with spatio-temporal trade-offs to improve coding efficiency. The proposed algorithm is performed at the object level for object-based video coding. Simulation results demonstrate moderate improvement at low as well as high bit rates. The proposed algorithm can produce the actual coded bits very close to the target bits over a wide range of bit rates. Consequently, the proposed algorithm has not experienced any buffer overflow or underflow over the bit rates between 32 kbps and 256 kbps.

Keywords: MPEG-4, Rate Control, Rate-Distortion, Object-based Coding

I. 서 론

H.263^[1], MPEG-1^[2], MPEG-2^[3], MPEG-4^[4]와 같은 비디오 부호화 표준들은 가변길이 부호화 (Variable Length Coding) 방법을 사용하기 때문에 비디오 부호화 과정을 통해 각 화면마다 발생하는 비트

수가 변한다. 이들 가변 비트열을 전송 대역폭이 고정된 채널을 통해 전송하고자 할 때, 보통 출력 비트열의 동적인 변동을 평활화하기 위해 채널 버퍼가 사용되며, 채널 버퍼의 넘침이나 결핍을 방지하기 위해 비트율 제어가 필요하다.

부호화되는 과정에서 매크로블록 양자화 값의 변화만을 통해 비트 발생량을 제어하는 MPEG-2 표준^[5]과는 달리, H.263^[6]과 MPEG-4 표준^[7]에서는 비트량을 제어할 수 있는 방법으로 화면건너뗂음을 지원한다. 따라서 부호기는 버퍼 제한사항을 만족시키거나 부호화 과정을 최적화하기 위해 화면건너뗂음을 선택한다. 부호화 과정에서 건너뛰어진 화면은 복호화 과정에서 이전에 복

* 정회원, LG전자기술원 정보기술연구소
(Information Technology Laboratory, LG Electronics Institute of Technology)

** 정회원, 광주과학기술원 정보통신공학과
(Department of Information and Communications, Kwangju Institute of Science and Technology)
접수일자 : 2003년 2월 11일, 수정완료일 : 2004년 1월 7일

원된 화면을 그대로 다시 보여주기 때문에, 공간적 화질 이외에 시간적 화질도 부호화 성능을 측정하는 중요한 성분이 된다.

현재까지 제안된 대부분의 알고리즘에서는 화면의 건너뛰기 단지 버퍼 상태에 따라 결정되기 때문에, 복원 후에 재구성된 건너뛰어진 화면들은 화질의 열화를 초래하게 된다. 따라서 버퍼의 상태뿐만 아니라 화면의 특성을 고려하여 부호화할 수 있는 알고리즘이 요구된다. 화면의 건너뛰기를 고려한 알고리즘들도 제안되었지만^[6,8,9], 이 알고리즘들은 모두 고정된 화면건너뛰기 비율(frameskip)만을 지원한다. 공간적 화질과 시간적 화질 사이의 절충을 고려하여 가변 화면건너뛰기 비율을 지원하는 알고리즘이 제안되었지만^[10], 최적의 경로를 찾기 위해 기본적으로 다양한 Q값에 대해 반복적으로 부호화 과정을 수행한 후에 최적의 화면건너뛰기 비율과 양자화 변수를 정하기 때문에 연산복잡도가 대단히 높다.

최근에 A. Vetro에 의해 부호화 화면과 건너뛰어진 화면의 왜곡을 예측할 수 있는 알고리즘^[11]과 비트율 제어 알고리즘^[12]이 제안되었다. 이 방법은 공간적 화질과 시간적 화질 사이의 절충을 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 화면건너뛰기 비율을 계산하기 위해 정해진 최대값 내에 있는 모든 값에 대해 왜곡을 계산하기 때문에 많은 연산이 요구된다.

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

본 논문의 구성은 다음과 같다. II장에서는 부호화 객체와 건너뛰기 객체의 왜곡을 예측하는 모델을 제시한다. III장에서는 제시한 예측 모델을 기반으로 부호화 성능을 향상시키기 위한 두 개의 비트율-왜곡 모델을 제안하고, IV장에서는 제안된 모델을 이용한 비트율 제어 알고리즘을 제안한다. V장에서 컴퓨터 실험 결과를 통해 제안된 알고리즘의 성능을 평가하고, 마지막으로 VI장에서 결론을 내린다.

II. 비트율-왜곡 모델

부호화 객체의 왜곡은 양자화 변수 값에 의해 결정되는 반면에, 건너뛰기 객체의 왜곡은 양자화 변수뿐만 아니라 화면건너뛰기 비율(frameskip rate, f_s)에 의해 결정된다. 본 논문에서는 건너뛰기 객체를 복호기에서 복원될 때, 이전에 부호화된 객체를 다시 재현하는 것을 기본으로 한다. 그러므로 건너뛰기 객체의 왜곡은 이전 부호화된 객체의 양자화 변수 값에 직접적으로 영향을 받는다. i 번째 화면에 있는 모든 객체들이 부호화되고, $(f_s - 1)$ 화면이 건너뛰어진 후에 $(i + f_s - 1)$ 번째 화면이 부호화되는 경우, 이 시간 동안의 부호화 왜곡은 다음과 같이 계산된다.

$$\bar{D}_{(i, f_s)}(Q_{i+f_s}, f_s) = \frac{1}{f_s} \left[D_c(Q_{i+f_s}) + \sum_{k=i+1}^{i+f_s-1} D_s(Q_i, k) \right] \quad (1)$$

1. 부호화 객체 왜곡

영상 부호화 방식에서 시간 $t = t_i$ 에 부호화되는 j 번째 객체의 왜곡은 다음과 같이 표현될 수 있다^[11].

$$D_c(Q_{j,i}) = a \cdot 2^{-2R_j(t_i)} \cdot \sigma_{z_{j,i}}^2 \quad (2)$$

여기서 $\sigma_{z_{j,i}}^2$ 와 $R_j(t_i)$ 는 i 번째 시간, j 번째 객체의 분산 값과 화면 내에 있는 화소당 평균 비트율이고, $Q_{j,i}$ 는 양자화 변수, a 는 입력 신호와 양자화 특성에 의해 결정되는 상수이다. 따라서 M 개의 객체를 포함하는 i 번째 화면의 부호화 왜곡은 다음과 같이 계산된다.

$$D_c(Q_i) = \sum_{j=0}^{M-1} D_c(Q_{j,i}) \quad (3)$$

2. 건너뛰기 객체 왜곡

i 번째 화면이 부호화되고, 그 다음 k 번째 건너뛰어진 화면내의 j 번째 객체의 왜곡은 다음과 같다^[11,13].

$$D_s(Q_{j,i}, k) = D_c(Q_{j,i}) + E_j \{ \Delta^2 z_{i,k} \} \quad (4)$$

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

$$E_j \{ \Delta^2 z_{i,k} \} = \sigma_{x_{j,i}}^2 \sigma_{\Delta x_{j,i}}^2 + \sigma_{y_{j,i}}^2 \sigma_{\Delta y_{j,i}}^2 \quad (5)$$

여기서 $\sigma_{x,j}^2$ 와 $\sigma_{y,j}^2$ 는 i 번째 화면에 있는 j 번째 객체의 x 방향과 y 방향의 기울기 값들의 분산 값을 나타내고, $\sigma_{v_x,j}^2$ 와 $\sigma_{v_y,j}^2$ 는 x 방향과 y 방향의 움직임 벡터들의 분산 값을 나타낸다.

MPEG-4 표준에서 다중 객체 부호화를 지원하기 위해서는 식 (2)와 식 (4)의 모든 값들은 객체가 정의되는 화소 내에서만 계산되어야 한다. 특히, 식 (2)에서 화소당 평균 비트율은 모양 정보와 움직임 정보를 포함하는 헤더 정보를 고려해서 계산되어야 한다. i 번째 화면이 부호화되고, 그 다음 k 번째 건너뛰어진 화면의 왜곡은 다음과 같이 표현된다.

$$D_s(Q_i, k) = \sum_{j=0}^{M-1} D_s(Q_{j,i}, k) \quad (6)$$

3. 모델 정확도

컴퓨터 실험을 통해 AKIYO와 같은 움직임이 적은 영상에 대해서는 실제 발생된 왜곡과 예측에 의해 계산된 왜곡과의 차이가 별로 없었지만, 움직임이 많은 영상에 대해서는 약간의 차이를 보인다.

하지만, 이 경우 최적화된 부호화기에서는 화면을 건너뛰는 것보다는 화면을 부호화하여 시간적 해상도를 높이는 것을 선호하기 때문에, 움직임이 많은 영상에 대해서도 제안된 모델을 적용하는 것이 가능하다. 또한 움직임 예측 과정이 정확하게 수행된다면 객체의 움직임이 많더라도 움직임 보상을 수행한 뒤에 얻은 나머지(residual) 영상은 큰 변동이 없기 때문에 제안한 모델을 적용할 수 있다.

III. 비트율-왜곡 부호화 모드

영상 부호화기의 성능을 향상시키기 위해 앞에서 제안된 R-D 부호화 모드를 이용하여 다음 화면을 부호화하는 것이 좋은지, 건너뛰는 것이 좋은지를 결정한다. 본 논문에서 제안된 알고리즘은 고정된 간격으로 화면을 건너뛰면서 부호화하는 대신에, 지정된 시간간격 내에서 가변 간격으로 화면을 부호화하는 것을 지원한다.

그림 1은 화면의 부호화 모드와 할당된 목표 비트율을 나타낸다. 그림 1에서 f_s 는 현재 화면에 할당된 비트량이 이전에 부호화된 화면의 헤더 정보, 움직임 정보, 모양 정보를 위해 필요한 비트량보다 적은 경우에 ($f_s -$

1) 화면이 강제로 건너뛰어져야 하는 것을 나타낸다. 그림 1에서 제안된 부호화 모드는 주어진 버퍼의 제한 내에서 연속된 두 화면을 부호화하는 것(경우 A)이 부호화 효율이 좋은지, 한 화면을 건너뛰고 다음 화면을 부호화하는 것(경우 B)이 부호화 효율이 좋은지를 결정한다.

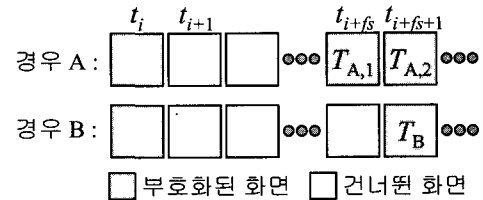


그림 1. 부호화 모드 (모델 I)
Fig. 1. Coding Mode (Model I)

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

$$\bar{D}_{i,A}^j(Q, f_s) = \sum_{k=i+1}^{i+f_s-1} D_s(Q_{j,i}, k) + D_c(Q_{j,i+f_s}) + D_c(Q_{j,i+f_s+1}) \quad (7)$$

$$\bar{D}_{i,B}^j(Q, f_s) = \sum_{k=i+1}^{i+f_s-1} D_s(Q_{j,i}, k) + D_s(Q_{j,i}, i+f_s) + D_c(Q_{j,i+f_s+1}) \quad (8)$$

여기서 첫 번째 항목은 이전에 부호화된 화면의 헤더 정보 때문에 건너뛰어진 ($f_s - 1$) 화면의 건너뛰어 왜곡을 나타낸다. 만약 입력 영상을 높은 비트율로 부호화하여 현재 화면에 할당된 비트량이 충분한 경우에는 이 항목은 사라진다. 또한, 경우 A와 경우 B에 대한 부호화 모델 모두 세 번째 항목에 같은 식을 가지지만, 식 (8)의 항목이 더 적은 왜곡을 가진다는 것을 짐작할 수 있다.

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 (7)의 두 번째 부호화 화면 왜곡을 계산하는 $D_c(Q_{j,i+f_s+1})$ 를 $D_s(Q_{j,i+f_s}, i+f_s+1)$ 로 변경하여 경우 A의 전체 부호화 왜곡을 계산한다.

객체기반 부호화에 대해 부호화 왜곡은 다음과 같이 정의된다.

$$d_{I,A}(Q, f_s) = \sum_{j=0}^{M-1} \bar{D}_{I,A}^j(Q, f_s) \quad (9)$$

$$d_{I,B}(Q, f_s) = \sum_{j=0}^{M-1} \bar{D}_{I,B}^j(Q, f_s) \quad (10)$$

여기서 M 은 부호화 객체의 수를 나타낸다. 모델 I에 대해 최적의 부호화 모드는 다음의 식에 의해서 결정된다.

$$\min\{d_{I,A}(Q, f_s), d_{I,B}(Q, f_s)\} \quad (11)$$

만약 화면과 화면 사이의 화질의 변화가 급격하게 변하는 것을 허용한다면, 위에서 제안한 부호화 왜곡 모델과 다음 장에서 제시하는 비트율 제어 알고리즘을 이용하여 현재 비디오 객체들의 양자화 변수를 정하면 된다. 하지만 화면과 화면 사이의 급격한 화질의 변화는 사람의 시각에 민감하게 반응하므로, 제안된 왜곡 모델에 화질의 급격한 열화를 방지할 수 있는 방법이 필요하다. MPEG-4 표준의 VM5 비트율 알고리즘 [7]의 경우, 현재 화면의 양자화 변수 Q 는 이전에 부호화된 화면의 양자화 변수 Q 에 대해 다음과 같은 범위 내에서 결정된다.

$$Q' - \frac{1}{4}Q' \leq Q \leq Q' + \frac{1}{4}Q' \quad (12)$$

본 논문에서 제안된 알고리즘에서도 현재 부호화 객체의 양자화 변수는 식 (12)의 제한 범위 내에서 정해지며, 양자화 변수의 제한 때문에 경우 B에서 경우 A와 비교하여 항상 2배의 성능을 얻지 못한다. 즉, 경우 A에서 첫 번째 부호화 화면에 할당된 비트수 T_1 과 두 번째 부호화 화면에 할당되는 비트수 T_2 의 합은 경우 B에서 부호화 화면에 할당되는 비트수 T_B 와 항상 같지 않다.

부호화 효율을 높이기 위해 할당된 비트량에 대한 양자화 변수가 큰 경우를 고려해 보자. 이 경우에 부호화기는 주어진 양자화 변수를 이용하여 현재 화면을 그대로 부호화할 수도 있지만, 현재 화면을 포함한 몇 개의 화면을 건너뛰으로써 생겨나는 잉여 비트량을 이용하여 그 다음 부호화 화면에 더 많은 비트량을 할당하는 것이 비트율-왜곡 측면에서 더 나은 성능을 제공할 수 있다. 이 경우에 비트량의 부족으로 인해 그 다음에 부호화되는 화면의 왜곡은 그 뒤 화면의 왜곡에도 영향을 미친다.

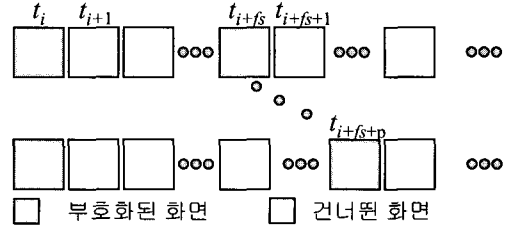


그림 2. 부호화 모드 (모델 II)

Fig. 2. Coding Mode (Model II)

그림 2는 양자화 변수가 큰 경우에 부호화 모드를 나타낸다. 이 경우에 전체 부호화 왜곡은 다음과 같이 정의된다.

$$\begin{aligned} \bar{D}_{II}^j(Q, f_s + p) = & \sum_{k=i+1}^{i+f_s+p-1} D_s(Q_{j,i}, k) + D_c(Q_{j,i+f_s+p}) \\ & + \sum_{k=i+f_s+p+1}^{i+f_{\max}} D_s(Q_{j,i+f_s+p}, k) \end{aligned} \quad (13)$$

여기서 $i+f_s+p$ 는 부호화되는 화면의 시간을 가리킨다. 식 (13)에서 마지막 항목은 바로 이전 화면을 부호화함으로써 건너뛰어져야 하는 화면의 왜곡을 가리킨다. 이때 f_{\max} 는 할당된 목표 비트량으로 현재 화면이 부호화된다고 가정하는 경우, 모양정보를 부호화한 비트량에 의해 강제로 건너뛰어지는 화면의 수로 결정된다.

객체기반 부호화에 대해 부호화 왜곡은 다음과 같이 정의된다.

$$d_{II}(Q, f_s) = \sum_{j=0}^{M-1} \bar{D}_{II}^j(Q, f_s + p) \quad (14)$$

모델 II에 대해 최적의 부호화 모드는 다음의 식에 의해 결정된다.

$$\arg \min d_{II}(Q, f_s) \quad (15)$$

기본적으로 본 논문에서 제안된 알고리즘은 해당 객체를 부호화한 경우와 건너뛴 경우에 해당 객체의 MSE 값을 예측하는 모델을 기반으로 현재 화면을 부호화할 지, 건너뛰고 그 다음 화면에 좀더 많은 비트를 할당할 것인지 정한다.

본 논문에서 제안된 알고리즘은 공간적-시간적 화질 절충을 부호화기가 부호화 과정에서 자동으로 조절할 수 있다. 즉, 연속하는 2개의 화면을 모두 부호화하는 것이 좋은지 현재 화면을 건너뛰고 다음 화면을 부호화하여 시간적 화질을 낮추고, 공간적 화질을 높이는 것이 좋은지 부호화기가 사용자의 간섭 없이 자동으로

결정한다. 기본적으로 모델 II의 경우에는 부호화기가 공간적-시간적 화질 절충을 조절하면서도, 건너뛰어지는 화면 개수의 최대값을 사용자가 정하기 때문에 시간적 화질이 떨어지는 것을 방지할 수 있다.

IV. 비트율 제어 알고리즘

1. 버퍼 제한사항과 부호화 모드 결정

앞에서 기술한 모델 I에 대해 발생 비트량은 다음의 조건을 만족해야 한다.

$$\bar{R} \leq R \tag{16}$$

이때 버퍼의 제한사항은 다음과 같다.

$$C_{I,A,f_s} \equiv \begin{cases} B_i + R(t_{i+f_s}) < B_{\max} \\ B_i + R(t_{i+f_s}) - f_s \cdot R_{\text{drain}} > 0 \end{cases} \tag{17}$$

$$C_{I,A,f_s+1} \equiv \begin{cases} B_{i+f_s} + R(t_{i+f_s+1}) < B_{\max} \\ B_{i+f_s} + R(t_{i+f_s+1}) - (f_s + 1) \cdot R_{\text{drain}} > 0 \end{cases} \tag{18}$$

$$C_{I,B,f_s} \equiv \begin{cases} B_i < B_{\max} \\ B_i - f_s \cdot R_{\text{drain}} > 0 \end{cases} \tag{19}$$

$$C_{I,B,f_s+1} \equiv \begin{cases} B_i + R(t_{i+f_s+1}) < B_{\max} \\ B_i + R(t_{i+f_s+1}) - (f_s + 1) \cdot R_{\text{drain}} > 0 \end{cases} \tag{20}$$

여기서 R 은 목표 비트율, B_{\max} 는 버퍼의 크기, B_i 는 현재 버퍼의 상태, R_{drain} 은 화면당 버퍼를 빠져나가는 양을 나타낸다.

모델 II에 대해 비트량 제한사항은 식 (16)을 따르고, 버퍼의 제한사항은 다음과 같다.

$$C_{II,p-1} \equiv \begin{cases} B_i < B_{\max} \\ B_i - (f_s + p - 1) \cdot R_{\text{drain}} > 0 \end{cases} \tag{21}$$

$$C_{II,p} \equiv \begin{cases} B_i + R(t_{i+f_s+p}) < B_{\max} \\ B_i + R(t_{i+f_s+p}) - (f_s + p) \cdot R_{\text{drain}} > 0 \end{cases} \tag{22}$$

비트율을 제어하면서 현재 화면에 타당한 부호화 모드를 결정하기 위해 그림 3에 보인 알고리즘을 이용한다. 본 논문에서 제안된 알고리즘의 비트율 제어 과정에서 기술되지 않은 부분은 MPEG-4 표준의 VM5 알고리즘에서 구현된 알고리즘을 이용한다. 특히, 모양 정보를 부호화하는 과정에서의 비트율 제어는 현재 남아있는 비트량과 이전에 건너뛰어진 화면의 수에 따라 부표본화(sub-sampling) 과정이 필요하지 결정한다.

1. f_s 값을 1로 설정하고, 현재 화면에 할당 가능한 비트수(T_V)를 계산한다.
2. 현재 버퍼 상태를 고려하여 초기 할당 비트수(T_B)를 결정한다. 조절된 비트수(T_B)가 이전 화면의 헤더 정보를 부호화하는데 사용된 비트수보다 적은 경우에는, 현재 할당된 비트수로는 현재 화면을 부호화하는데 적절하지 않으므로 f_s 의 값을 증가시키면서 다시 조절한다.
3. 화면에 할당된 비트수를 각 객체의 특성에 알맞게 분배하고, 각 객체에 할당된 비트수와 이전 해당 객체의 양자화 변수(QP)를 고려하여 현재 객체의 QP 값을 결정한다.
4. 경우 A와 경우 B에 대한 부호화 왜곡을 계산하고, 식 (17)에서 식 (20)까지의 버퍼 조건을 만족하는지 검사한다. 버퍼 조건을 만족할 때까지 f_s 값을 증가시키면서 현재까지의 과정을 반복한다.
5. 결정된 QP 값이 큰 경우에, 부호화 왜곡을 예측하여 p의 값을 구한다. 현재 QP 값이 큰지 작은지 구분하는 임계값을 정확하게 결정하는 것은 어려운 문제이다. 본 논문에서는 실험을 통해 26을 임계값으로 선택하였다.
6. 화면의 부호화 과정을 수행한다.

그림 3. 비트율 제어 알고리즘
Fig. 3. Rate Control Algorithm

2. 목표 비트량 할당

현재 화면의 목표 비트량은 남아있는 화면의 수와 버퍼의 상태를 이용하여 결정된다^[7]. 이때 결정된 목표 비트량은 현재의 버퍼 상태에 따라 다음과 같이 조절된다.

$$T_B = T_V \cdot \frac{\tilde{B}_i + 2(B_{\max} - \tilde{B}_i)}{2\tilde{B}_i + (B_{\max} - \tilde{B}_i)} \tag{23}$$

여기서 T_V 는 초기 목표 비트량으로 현재 남아있는 비트수 T_r , 이전 화면을 부호화하는데 사용된 비트수 T_p , 남아있는 화면의 개수 N_r 에 따라 결정된다. 수정된 버퍼 총만도 \tilde{B}_i 는 다음과 같다.

$$\tilde{B}_i = B_i - (f_s - 1) \cdot R_{\text{drain}} \tag{24}$$

여기서 f_s 는 부호화 모드에 의해 건너뛰어야 하는 화면의 수이다.

3. 목표 비트량 분배

한 화면에 속하는 각각의 객체는 각기 다른 양자화 변수를 이용하여 부호화할 수 있기 때문에, 각 화면에 할당된 목표 비트량은 각 비디오 객체의 특성에 맞게 알맞게 분배되어야 한다. 현재 제안된 알고리즘들은 주

어진 비디오 객체의 왜곡을 예측하기 위해, 객체의 크기, 움직임 정보, 분산 값을 이용하여 각 객체에 적절히 비트량을 분배하였다^[7]. 하지만 이 방법들은 T_B 가 이전에 부호화된 화면의 헤더 정보(T_{hdr})보다 충분히 큰 경우에도 각 객체의 텍스처 정보를 부호화하는데 할당되는 목표 비트량은 이전 부호화 시간에서 모양 정보를 부호화하기 위해 사용된 비트량보다도 적게 할당할 수 있다. 이 문제를 해결하기 위해 본 논문에서는 다음과 같이 각 객체에 비트량을 할당하였다.

$$T_j = (T_B - T_{hdr}) \cdot (w_m \text{MOT}_j + w_v \text{VAR}_j) + T_{hdr,j} \quad (25)$$

여기서 $T_{hdr,j}$ 는 이전에 부호화된 화면에서 j 번째 객체의 헤더 정보, 움직임 정보, 모양 정보를 표현하기 위해 사용된 비트량이고, T_{hdr} 는 모든 객체에 대해 $T_{hdr,j}$ 의 합이다. MOT_j 와 VAR_j 는 각각 모든 객체의 움직임을 나타내는 MOT, 분산 값을 나타내는 VAR에 의해 정규화된 j 번째 객체의 움직임 값과 분산 값이다. 가중치 성분 w_m 과 w_v 는 모두 0과 1 사이에 존재하고, 두 성분의 합은 1이 되어야 한다.

4. 산술적 복잡도

부호화 객체와 건너뛴 객체의 왜곡을 계산하기 위해 실제 부호화 과정을 거친 후에 이 값을 결정하게 되면 정확한 값을 얻을 수 있다는 장점을 가지지만, 산술적 복잡도가 매우 높고 부호화 지연이 발생하는 문제점이 있다.

부호화 지연이 없으면서 산술적 복잡도를 줄이기 위해 본 논문에서는 이전에 부호화된 정보들을 이용하여 객체의 왜곡을 계산한다. 부호화 객체의 왜곡을 계산하기 위해서 해당 객체의 분산 값을 결정해야 하는데, 분산 값은 매 화면마다 급격히 변하지 않고 거의 일정한 값을 유지하는 특성이 있기 때문에^[14], 이전 화면을 부호화한 후에 결정된 값을 그대로 이용한다. 또한, 실제 텍스처 정보를 부호화하는데 할당되는 비트량은 현재 객체에 할당된 목표 비트량에서 이전 시간에서 모양정보를 부호화한 후에 발생한 비트량, 움직임 정보를 부호화한 후에 발생한 비트량, 헤더 정보를 표현하기 위해 발생한 비트량을 뺀 값을 이용한다. 이 과정에서 모양 정보에 대한 정확한 예측 방법과 움직임 벡터를 표현하기 위해 사용된 비트량을 정확하게 예측하는 방법이 이용된다면 좀더 정확한 예측 모델을 만들 수 있다.

건너뛴 객체의 왜곡을 계산할 때, k 번째 부호화 시간에 있는 j 번째 객체의 움직임 벡터 분산 값은 매 화면

마다 객체는 급격히 변하지 않고 일정하게 변한다고 가정하여 다음과 같은 식을 이용하여 구한다.

$$\sigma_{\Delta_{j,i}}^2 \approx \sigma_{\Delta_{j,i-n}}^2 \cdot \left(\frac{k-i}{f_i} \right)^2 \quad (26)$$

여기서 f_i 는 이전에 부호화된 화면과 그 참조 화면 사이의 간격을 나타낸다.

본 논문에서는 기울기 값들의 분산 값을 결정하기 위해 Prewitt 알고리즘을 사용하여 해당 화소 값의 기울기를 계산하였다. 분산 값들이 결정된 후에 부호화 왜곡과 건너뛴 왜곡을 계산하는 과정은 간단하기 때문에 연산 복잡도가 높지 않다. 또한, 본 논문에서 제안된 알고리즘은 기존의 방법에 비해 부호화되는 화면의 개수가 적기 때문에 실제 부호화 시간은 기존의 방법에 비해 현저히 감소한다.

V. 실험 결과 및 분석

본 논문에서 제안된 알고리즘의 성능을 검사하기 위해 본 논문은 2개의 비디오 객체를 포함하면서 300 화면으로 구성된 CIF 형식의 AKIYO 영상과 FOREMAN 영상을 이용하였다. 영상의 부호화 성능을 비교하기 위해 MPEG-4 참조 알고리즘(VM5)^[15]과 제안된 알고리즘을 이용하여 32 kbps에서 256 kbps의 비트율로 부호화한 결과를 구하였다.

본 논문에서 제안된 알고리즘의 성능을 평가하기 위해, 부호화 과정에서 목표로 하는 비트율에 정확하게 실제 비트량이 발생하는지를 검사하고, 비트율 제어 방법에서 우선적으로 요구되는 버퍼 넘침 현상과 버퍼 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 4는 32kbps에서 256 kbps까지 설정된 목표 비트율로 MPEG-4 VM5 알고리즘과 본 논문에서 제안된 알고리즘으로 부호화한 후에 목표 비트율과 실제 부호화 비트율 사이의 관계를 나타낸 것이다. 그림 4에 보인 것처럼, 기존의 알고리즘은 모든 비트율에서 목표 비트율보다 많은 비트량을 발생시키는 것에 반해, 본 논문에서 제안된 알고리즘은 목표 비트율에 거의 비슷하게 발생하는 것을 확인하였다.

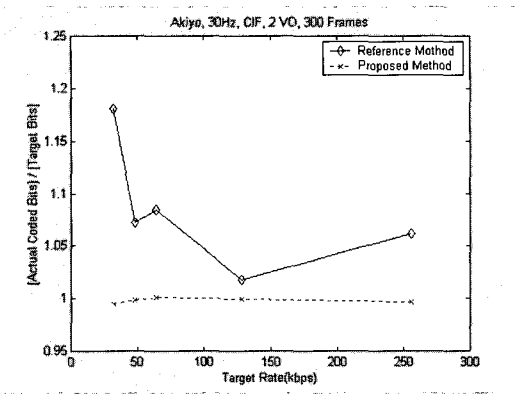
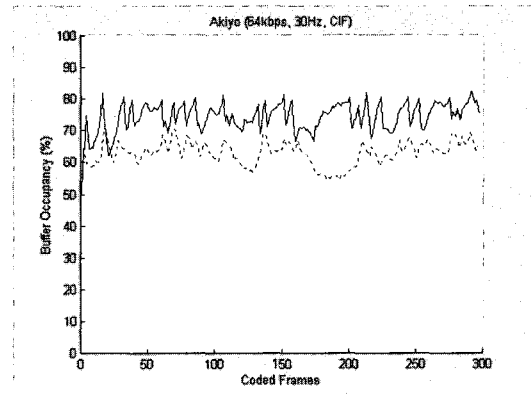
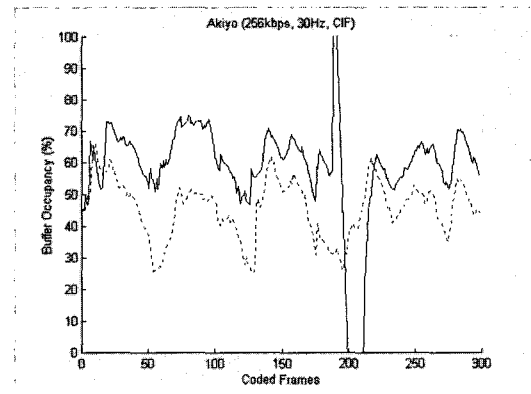


그림 4. 목표 비트량과 실제 비트 발생량 비교 (실선 : VM5 알고리즘, 점선 : 제안된 알고리즘)
 Fig. 4. Comparison of Target Bits and Actual Coded Bits (Solid Line : VM5 Algorithm, Dotted Line : Proposed Algorithm)

그림 5는 각각의 비트율에서 버퍼의 상태를 나타낸 것이다. 그림 5에서 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. 실험 결과 그림으로 보이지는 않았지만, 실험된 다른 비트율에서도 비슷한 성능을 보이는 것을 확인하였다.



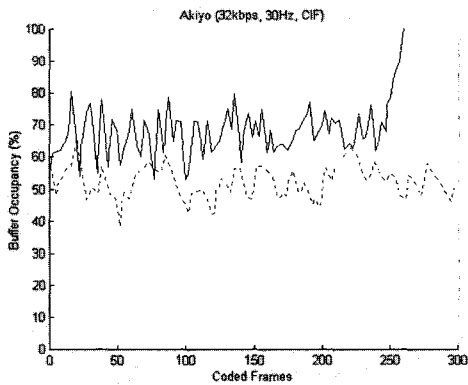
(b) 64 kbps



(c) 256 kbps

그림 5. 버퍼 상태 비교 (실선 : VM5 알고리즘, 점선 : 제안된 알고리즘)

Fig. 5. Comparisons of Buffer Occupancy (Solid Line : VM5 Algorithm, Dotted Line : Proposed Algorithm)



(a) 32 kbps

표 1은 실험된 비트율에서 MPEG-4 VM5 알고리즘과 제안된 알고리즘의 PSNR 결과를 비교한 것이다. 표 1에서 알 수 있듯이, 제안된 알고리즘이 기존의 방법에 비해 0.5dB에서 1dB 정도의 성능 향상을 보이는 것을 알 수 있다. 기존의 알고리즘은 단지 버퍼의 상태에 의해서만 화면이 건너뛰어지지만, 제안된 알고리즘에서는 버퍼 상태뿐만 아니라 최소 왜곡 모델을 이용하기 때

표 1. PSNR 결과 비교

Table 1. Comparisons of PSNR Values

시퀀스	객체	VM5 알고리즘 (kbps)					제안알고리즘 (kbps)				
		32	48	64	128	256	32	48	64	128	256
AKIYO	0	39.13	39.15	39.23	41.87	45.88	39.24	39.23	39.50	42.60	45.71
	1	29.12	29.65	30.53	34.37	37.10	30.11	30.31	30.55	34.47	37.55
FOREMAN	0	32.21	30.97	31.12	31.14	31.05	33.00	31.88	31.45	31.20	31.16
	1	33.20	30.85	30.40	30.10	30.49	34.87	32.92	31.53	30.42	30.66

문에 더 나은 성능을 제공한다. 성능 향상이 미미한 비트율에서도 제안된 알고리즘은 버퍼 넘침이나 버퍼 결핍을 발생하지 않지만, 기존의 방법은 버퍼 넘침이나 버퍼 결핍 현상이 종종 발생한다.

한 화면을 구성하는 모든 객체를 부호화하는 경우에 모든 객체를 하나의 화면으로 구성하여 화면 기반으로 시퀀스를 부호화하는 것이 좋은지, 각 객체별로 별도로 부호화하는 것이 좋은지 결정하는 것은 성능 평가에 있어서 중요한 관건이다. 제안된 알고리즘을 그대로 화면기반 부호화 방식에 적용하게 되면, 객체의 모양 정보를 부호화하는데 걸리는 시간과 목표 비트량 분배와 같은 과정이 필요하지 않기 때문에 기존의 방법에 비해 부호화 효율은 높이면서 부호화 시간은 줄어든다. 하지만 객체기반 부호화 방식은 배경과 같은 움직임이 적은 객체에 적은 비트량을 할당하고, 움직임이 많은 객체에 많은 비트량을 할당함으로써 부호화 효율을 극대화할 수 있을 뿐만 아니라, MPEG-4 참조 소프트웨어에서 지원하는 UPM 방식과 같은 비트율 제어 과정을 일부 지원할 수 있는 장점이 있다.

VI. 결 론

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 부호화 성능을 제공하며 안정된 버퍼 상태를 유지한다. MPEG-4 부호화 알고리즘은 객체기반 부호화를 지원하기 위해, 제안된 객체기반 왜곡 모델을 이용하여 부호화 성능을 향상시키는 알고리즘의 개발이 가능하다^[16].

참 고 문 헌

- [1] ITU-T. Recommendation H.263 Video coding for low bit rate communication, 1998.
- [2] ISO/IEC 11172-2, "Information technology Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s," Part 2:Video, 1993.
- [3] ISO/IEC 13818-2, "Information technology Generic coding of moving pictures and associated audio," Part 2:Visual, 1996.
- [4] ISO/IEC 14496-2, "Information technology Coding of audio/visual objects," Part 2: Visual, 1998.
- [5] MPEG-2 Video Test Model 5, ISO/IEC JTC/SC29/ WG11 MPEG93/457, Apr. 1993.
- [6] T. Weigand, M. Lightstone, D. Mukherjee, T. Campbell, and S. Mitra, "Rate-distortion optimized mode selection for very low bit-rate video coding and the emerging H.263 standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 182-190, Apr. 1996.
- [7] 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.
- [8] H. Sun, W. Kwok, M. Chien, and C. John, "MPEG coding performance improvement by jointly optimizing coding mode decision and rate control," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 449-458, June 1997.
- [9] J. Lee and B. Dickenson, "Rate-distortion optimized frame type selection for MPEG encoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 501-510, June 1997.

- [10] S. Liu, J.W. Kim and C.-C. Jay Kuo, "Bit allocation for video coding with temporal-spatial tradeoff," Proc. IEEE Pacific-Rim Conference on Multimedia (PCM'2001), Beijing, China, Oct. 2001.
- [11] 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.
- [12] A. Vetro, Y. Wang, and H. Sun, "Rate-distortion optimized video coding considering frameskip," *ICIP '01*, pp. 534-537, Oct. 2001.
- [13] 이정우, 호요성, "MPEG-4 비디오 부호화를 위한 객체기반 비트율 제어 방법," 전자공학회 컴퓨터 소사이어티 추계학술대회, pp. 143-146, Nov. 2001.
- [14] A. Vetro, "Object-based encoding and transcoding," Ph.D. Dissertation, Dept. Elect. Eng., Polytechnic Univ., Brooklyn, NY, June 2001.
- [15] ISO/IEC 14496-5, "Information technology Coding of audio/visual objects," July 2000.
- [16] J.W. Lee, A. Vetro, Y. Wang, and Y.S. Ho, "MPEG-4 video object-based rate allocation with variable temporal rates," *IEE Electronics Letters*, vol. 38, no. 19, pp. 1088-1090, Sept. 2002.

저 자 소 개



이 정 우(정회원)

1996. 2 : 전북대학교 정보통신공학과 졸업(학사), 1998. 2 : 광주과학기술원 정보통신공학과 졸업(석사), 2003. 8 : 광주과학기술원 정보통신공학과 졸업(박사), 2003. 8 ~ 현재 : LG전자기술원 선임연구원, 소속기관: LG전자기술원 정보기술연구소, <주관심분야 : 영상신호 처리 및 압축, 영상통신, 비트율 제어, 디지털 데이터 방송>



호 요 성(정회원)

1981. 2 : 서울대학교 전자공학과 졸업(학사), 1983. 2 : 서울대학교 전자공학과 졸업(석사), 1983. 3 ~ 1995. 9 : 한국전자통신연구소 선임연구원, 1989. 12 : University of California, Santa Barbara, Dept. of Electrical and Computer Eng. 졸업(박사), 1990. 1 ~ 1993. 5 : 미국 Philips 연구소 Senior Research Member, 1995. 9 ~ 현재 : 광주과학기술원 정보통신공학과 교수, 소속기관: 광주과학기술원 정보통신공학과, <주관심분야 : 디지털 신호처리, 영상신호처리 및 압축, 초저속 영상통신, 디지털 TV와 고선명 TV 방식, 멀티미디어 통신, 디지털 워터마킹, 3차원 실감방송 기술>