2차원 히스토리 풀과 슬라이딩 윈도우를 이용한 MPEG-4 비디오 프레임 기반 비트 생성율 제어 방법

(MPEG-4 Video Frame-based Bitrate Control using 2D History Pool and Sliding Window)

박 광 훈 † 이 윤 진 ††
(Gwang Hoon Park) (Yoon Jin Lee)

요 약 본 논문에서는 2차원 히스토리 풀과 슬라이딩 윈도우를 이용한 MPEG-4 비디오 프레임 기반 비트 생성율 제어 방법을 제안한다. 제안된 방법은 먼저 영상 특성에 따라 코딩 결과를 분류하여 2차원 히스토리 풀에 저장한다. 슬라이딩 윈도우를 이용하여 코딩하려는 영상 프레임의 특성과 비슷한 특성을 갖는 코딩 결과들을 풀에 저장된 결과로부터 수집한다. 마지막으로 수집된 코딩 결과를 이용하여 귀환회귀를 수행하는 방법이다. 제안된 방법을 이용하면 양자화 스텝 결정 시 외삽이 발생되는 경우를 줄여줌으로써 급변하는 영상 특성 변화에 적극적으로 대처할 수 있다. 제안된 비트 생성율 제어 방법은 기존의 MPEG-4 프레임 기반 비트 생성율 제어 방법에 비해 전송선로의 제약을 만족하는 프레임 당 발생된 비트량, 피크 신호 대 잡음비 그리고 프레임 스킵 수의 비교에 있어서 성능이 우수함을 확인하였다.

키워드 : MPEG, 비디오 코딩, 전송률 제어, 비트율 제어, 동영상 코딩

Abstract *This paper introduces the MPEG-4 video frame-based bitrate control methodology using two dimensional history pool and a sliding window. Proposed method preferentially clusters the encoded results according to the image characteristics and stores those results into the 2 dimensional history pool. Among the stored results in the pool, the sliding window collects the encoded results whose characteristics are very similar to the image frame to be encoded. Feedback regression is finally carried out based on the collected results. Therefore proposed method can actively adapt to the rapid varying image characteristics by reducing the occurrences of the extrapolations when determining the quantization steps. Proposed method has better performances than the MPEG-4 frame-based bitrate control algorithm by evaluating with the actually encoded bits per frame, encoded PSNR's, and frame skips.*

Key words : *MPEG, Video Coding, Rate Control, Bitrate Control*

1. 서 론

비디오 코딩에 있어 영상의 비트율 제어 방법으로는, 가변 비트율에서의 비트율 제어 방법(Variable Bit Rate: VBR)과 균일 비트율에서의 비트율 제어 방법(Constant Bit Rate: CBR) 등의 두 가지의 방법이 있다. 가변 비트율 제어 방법은 ATM 망에서와 같이 주어진 비트율에 대해 최적의 영상 품질을 얻기 위해 비트율을 가변적으로 제어하도록 설계되어 있으며[1], 균일 비트율 제어 방법은 버퍼의 제약을 만족하며 낮은 대기시간을 갖도록 비트율을 제어하도록 설계되어 있다. 주로 사용하는 균일 비트율 제어 방법은 현재 전송해야 하는 정보량과 이미 전송되어 보내진 비트, 그리고 현재 버퍼에 남아있는 정보량의 관계를 이용하여 처리하는 귀환회귀(Feedback regression) 방법이 주로 사용되며, 이는 MPEG 비트율 제어 방법으로도 사용된다[2~5, 8~13]. MPEG 비트율 제어 방법으로 사용되는 귀환회귀 방법은 과거의 코딩정보가 저장된 히스토리(History)를 이용하여 왜곡 비트율(Rate distortion:

† 종신회원 : 경희대학교 전자정보학부 컴퓨터공학전공 교수
ghpark@khu.ac.kr
†† 학생회원 : 경희대학교 전자정보학부 컴퓨터공학전공
rainmake@video.khu.ac.kr
논문접수 : 2002년 1월 16일
심사완료 : 2002년 4월 19일

RD) 모델을 구성하고 이를 통해 회귀를 수행한다[2~7]. RD 모델은 과거에 코딩 완료된 영상정보를 바탕으로 구성되며, 영상의 특성이 비슷하거나 시간적으로 서서히 변화는 경우에는 효과가 크다. 하지만, 영상의 특성이 급격하게 변하는 경우에 있어서는 과거 코딩정보를 바탕으로 하는 RD 모델이 맞지않아, 타겟 전송량과 부합하는 비트량이 발생된다. 따라서, 영상특징의 급격한 변화에 적극적인 대응을 위해서는 종래의 RD 모델을 이용한 귀환 회기 방법에 대한 보완이 요구된다.

본 논문에서는 영상 특성을 고려한 새로운 프레임 기반 비트율 제어 방법을 제안한다. 제안된 방법은 기존의 귀환 회기 방법의 비트율 제어가 1차원 히스토리 큐(1D history queue)를 이용하여 시간적(temporal) 순서에 의해 순차적(sequential)으로 코딩 결과를 저장하고 회귀를 수행하는 방법을 사용함으로써 영상 특징의 변화에 적극적으로 대처하지 못한 점을 인지하여 2차원 히스토리 풀(2D history pool)을 구성하여 영상 특성에 따라 코딩 결과를 분류하고 저장한 후, 코딩하려는 영상 특성과 비슷한 특성을 갖는 코딩 결과만을 슬라이딩 윈도우(Sliding Window)를 이용하여 선택하여 재배치한 다음, 회귀를 수행하여 양자화 스텝(Quantization step: Q) 결정시 외삽(Extrapolation)이 발생되는 경우를 줄여줌으로써 영상 특성의 변화에 적극적으로 대처하는 방법이다. 본 논문의 구성은 다음과 같다. 2장에서는 MPEG-4 프레임 기반 비디오 비트율 제어 방법이 간단히 소개되었고 3장에서는 본 논문에서 제안하는 프레임 기반 비트율 제어 방법을 소개하였으며 4장과 5장에서는 제안된 방법의 실험 결과 및 그에 따른 결론을 도출하였다.

2. MPEG-4 프레임 기반 비디오 비트율 제어 방법

MPEG-4 프레임 기반 비트율 제어 방법은 계층적 비트율 제어 기법(Scalable Rate Control: SCR)으로서 버퍼의 제약 및 낮은 지연시간을 갖는 균일 비트율 제어와 지연제약(Delay constraints)을 고려하지 않는 가변 비트율 제어를 복합적으로 고려하여 이를 모두 충족하도록 디자인 되어있으며, 10kbps~4Mbps 등의 비트율, QCIF와 CIF 등의 공간적 해상도, 그리고 5fps~30fps 등과 같은 시간적 해상도에 대하여 다양하게 적용 할 수 있다[2~4].

MPEG-4 프레임 기반 비디오 비트율 제어 방법은 귀환회귀 방법으로, 현재 코딩 할 영상에 대해 타겟 비트율(Target bitrate)을 정하고 과거에 코딩 완료된 정보가 저장되어 있는 히스토리를 이용하여 RD모델을 갱신한 후, 해당 영상에 대한 양자화 스텝을 예측하여 엔코딩을 수행함으로써 비트 생성율을 조절한다. 이때 양자화 스텝 예측에 이용되는 RD 모델은 왜곡 비트율 이론을 기반으로 하고 있다. RD모델에서 영상 품질이 높은 경우는 양자화 스텝의 값이 낮은 경우를 말하며, 발생 비트량은 지수 함수적인(exponential) 증가에 따른다[2~4, 8, 9].

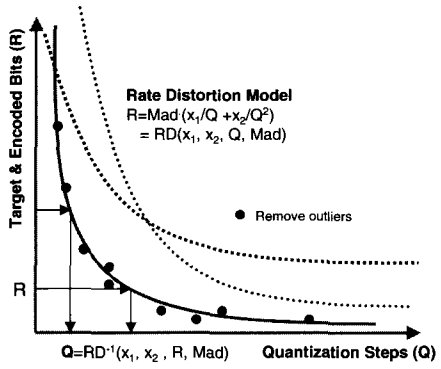


그림 1 왜곡비트율(RD) 모델

발생 비트량 R에 따른 양자화 스텝 Q의 예측 방법을 그림 1에 도식적으로 표현하고 있다. 그림 1에서 RD 모델은 양자화 스텝, 발생 비트량, 영상 복잡도(Mad)를 이용하여 다음과 같은 간단한 식으로 표현된다[3].

$$R = X_1 \, Mad / Q + X_2 \, Mad / Q^2 \qquad (1)$$

식 (1)에서 R은 영상에 대한 발생 비트량, Q는 양자화 스텝, Mad는 영상 복잡도(Mean Absolute Difference), 그리고 X_1과 X_2는 RD 모델 파라메터를 각각 나타낸다. 식 (1)에서 영상에 대한 양자화 스텝 예측시 영상 복잡도를 이용하는데, 본 논문에서 Mad의 의미는 원 영상(Original Image)과 움직임 추정 및 보상(Motion estimation / compensation)이 끝난 재생영상(Reconstructed Image)간의 차이 값의 절대치로서 엔코딩시 발생되는 비트 생성량과 밀접한 관계를 갖는다[2~3].

그림 1에서 표현된 바와 같이 MPEG-4 프레임 기반 비트율 제어는 프레임에 대한 타겟 비트량을 계산한 후, 이를 RD모델식을 이용하여 해당 영상 프레임의 Q값을 예측하고, 결정된 Q값을 이용하여 전송하고자 하는 영상 프레임의 엔코딩을 수행하는 과정으로 이루어져 있다. 이를 위하여, 이전 영상에 대한 코딩 결과로 발생된 {Q, *Encoded Bits*(실제 발생 비트량), Mad}정보를 사용하여 회귀를 수행하여 RD모델 파라메터 X_1과 X_2를

갱신한 후 코딩하려는 영상 프레임에 대한 Q값 예측을 수행한다. 이는 과거 영상 프레임의 특성을 바탕으로 엔코딩 하고자 하는 영상 프레임의 특성을 예측하기 위함이다.

프레임에 대한 타겟 비트량은 현재 버퍼 상태와 앞으로 엔코딩 해야 할 프레임 수, 그에 따른 여분의 비트율을 고려하여 결정되며, 이전 프레임에 대한 엔코딩 비트량이 많다면 현재 프레임에 대한 타겟 비트량은 이전 타겟 비트량보다 적은 양의 비트량으로 결정되고, 이전 프레임에 대한 엔코딩 비트량이 적다면 타겟 비트량은 이전 타겟 비트량보다 많게 결정된다. 결정된 타겟 비트량은 버퍼 상태를 고려하여 버퍼의 상한(upper bound)과 하한(lower bound)을 넘지 않도록 하여 버퍼의 오버플로우(overflow)와 언더플로우(underflow)가 발생하지 않도록 재조정된다[2].

결정된 타겟 비트량을 이용하여 RD 모델식을 통해 예측된 Q값은 이전 프레임의 Q값을 참고로 보정과정을 거친다. Q값에 대한 보정은 RD 모델을 바탕으로 예측된 양자화 스텝이 이전 프레임의 양자화 스텝과 비슷한 범위의 값을 갖도록 하여 영상 품질에 대한 연속성을 유지하도록 한다[2~3]. 엔코딩 수행 후에 버퍼의 사용량이 버퍼의 80% 이상이 되는 경우에는 다음 프레임을 스킵(skip)하여 버퍼의 오버플로우를 방지한다. 그리고, 엔코딩 결과로 나온 프레임 당 영상 특성정보 {*Mad, Encoded Bit, Q*}는 그림 2에서 도시한 바와 같이 시간축을 기준으로 하는 순차적 FCFS(First Come First Serve) 1차원 히스토리 큐에 저장되며, 이들은 RD 모델을 갱신하고 다음 프레임에 대한 양자화 스텝에 대한 예측에 활용된다.

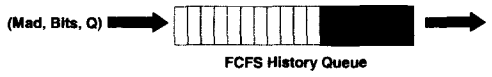


그림 2 1차원 FCFS 히스토리 큐

MPEG-4 프레임 기반 비트율 제어 방법을 그림 3에 도시하고 있다[2]. MPEG-4 프레임 기반 비트율 제어 방법은 사전 엔코딩(Pre-Encoding)과정과 엔코딩 과정, 그리고 사후 엔코딩(Post-Encoding)과정으로 나뉘어 진행된다. 사전 엔코딩 과정에서는 비트율 제어의 초기화와 프레임에 대한 타겟 비트량을 설정하고 현재 프레임에 대한 Q값을 결정한다. 엔코딩 과정에서는 결정된 Q값을 이용하여 엔코딩을 수행하고 사후 엔코딩 과정에서 엔코딩 후 발생된 i번째 프레임 당 코딩 결과인 {*Mad*(i), *Encoded Bits*(i), Q(i)}를 히스토리에 저장하고 RD모델을 갱신한다. RD모델은 이미 FCFS 큐에 저장된 여러개의 *Encoded Bits*(i)를 식 (1)의 R값에 대입하고 또한 *Mad*(i)와 Q(i)를 식 (1)에 대입하여 RD 모델 파라메터 X_1과 X_2값을 도출해 내는 귀환 회기 방법을 사용하여 갱신된다.

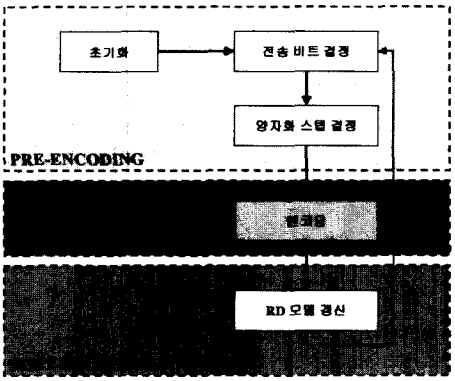


그림 3 MPEG-4 프레임 기반 비트율 제어 방법

MPEG-4 프레임 기반 비트율 제어 방법은 5단계로 구성되어 있으며 방법은 아래와 같다[2].

Step 1. 초기화

RD 모델 파라메터 X_1과 X_2를 초기화 한다.

Step 2. 프레임에 대한 타겟(전송) 비트율 계산

현재 프레임에 대한 타겟 비트량은 이전 프레임에 대한 발생 비트량과 버퍼 상태를 고려하여 계산된다. 타겟 비트량의 최소량은 최소 영상 품질을 얻을 수 있도록 해야 한다. 타겟 비트량은 버퍼 오버플로우나 언더플로우를 막기위해 버퍼 상태에 따라 조정되며, 현재의 버퍼 상태와 타겟 비트량의 합이 버퍼크기의 90% 이상인 경우에 오버플로우의 발생을 피하기 위해 타겟 비트량을 조정하고, 현재의 버퍼 상태와 타겟 비트량의 합에 기본 비트 전송량(R/N_f)을 뺀 비트량이 버퍼 크기의 10% 이하인 경우에 언더플로우의 발생을 피하기 위해 타겟 비트량을 조정한다. 이때 R/N_f는 영상 시퀀스에 대해 엔코딩에 사용할 수 있는 여분의 비트량을 엔코딩 해야 할 프레임 수로 나눈 비트량을 말한다.

T : 현재 프레임에 대한 타겟 비트량
T' : 이전 프레임의 엔코딩된 비트 발생량
R : 시퀀스에 대한 비트량에서 엔코딩 비트 발생량을 제하고 남은 여분의 비트량
N_f : 시퀀스에 대해 앞으로 엔코딩 해야 할 프레임 수
F : 타겟 비트율 (bits/second)
F_r: 프레임 율 (예: 25, 30)
a : 현재 버퍼량

```
b : 버퍼 크기에서 현재 버퍼량을 뺀 버퍼 여분량
    (BufferSize - a)
BufferSize : 버퍼의 크기, 보통 (F/2)으로 설정
past_percent : 상수 값

T=R/N_f * (1-past_percent)+T' * past_percent
T=(a + 2 * b)/(2 * a + b) * T
T=MAX(F/F_r, T)

if  (a+T > 0.9*BufferSize)
    T = MAX(F/ F_r, 0.9*b);          /* to avoid overflow */

else if  (a-R/N_f+T < 0.1*BufferSize)
    T = R/N_f - a + 0.1*BufferSize; /* to avoid underflow */
```

Step 3. 양자화 스텝(Q) 결정

타겟 비트량(T)과 Mad 그리고 이전 영상 프레임에 대한 RD모델 갱신 과정(Step 5)에서 갱신된 RD모델 파라메터 X_1과 X_2를 이용하여 현재 프레임에 대한 RD 모델을 구성한 후 양자화 스텝 Q를 결정한다. 양자화 스텝은 이전 영상 품질에 대해 일관성을 유지하기 위해서 Q'로 표시된 이전 프레임에 대한 양자화 스텝의 ±25% 이내의 값을 갖도록 제한하며, 1과 31사이의 값을 갖는다.

```
T : 현재 프레임에 대한 타겟 비트량
Mad : 현재 프레임에 대한 영상 복잡도
Q : 현재 프레임에 대한 양자화 스텝
Q' : 이전 프레임에 대한 양자화 스텝
MAX(a, b, c) : 세 개의 파라메터 a, b, c 중에 가장 큰 값을
               선택한다.
MIN(a, b, c) : 세 개의 파라메터 a, b, c 중에 가장 작은 값을
               선택한다.
T=X1 Mad/Q+X2 Mad/Q^2
Q=MIN(1.25 Q', Q, 31)
Q = MAX(0.75 Q', Q, 1)
```

Step 4. 현재 프레임에 대한 엔코딩

프레임단위의 엔코딩을 수행한다.

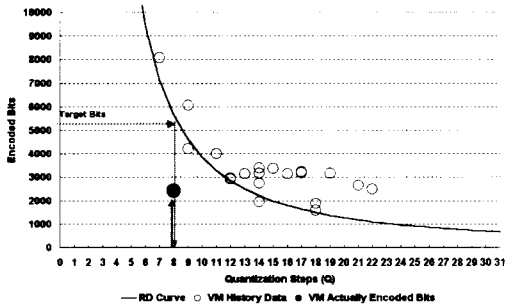
Step 5. RD모델 갱신

RD모델을 현재 프레임에 대한 코딩 결과를 기반하여 갱신한다. 이때, 프레임에 대한 헤더(header)와 움직임 벡터 전송에 사용된 비트량은 Q값 결정과 무관하기 때문에 갱신과정에서 제외한다. RD모델 갱신과정 중에서, 예측된 RD모델과 차이가 큰 과거의 코딩 결과 데이터를 히스토리로부터 제거한 후(Remove Outlier 과정) 다시 한번 회귀 과정을 거쳐 RD모델 파라메터 X_1과 X_2를 갱신한다. 또한 현재 버퍼의 80% 이상을 사용한 상태에 이르면 오버플로우를 방지하기 위해 다음 프레임을 스킵한다. 사전에 코딩 완료된 전송 결과를 1차원의 순차적 FCFS 큐에 저장하고 그들을 기반으로 RD 모델 파라메터 X_1과 X_2를 갱신 함으로서 행해지는 MPEG-4 비디오 비트율 제어 방법은, 코딩을 하려는 영상의 특성이 FCFS 큐에 저장된 코딩 결과와 매우 상이한 경우, 또는 비슷한 코딩 결과가 과거에 존재하였지만 FCFS 큐의 크기가 N개(약 20개)로 제한되어 해당 프레임의 영상 특성과 비슷한 결과를 저장하지 못하고 있는 경우에 있어서, 현재의 영상 특성과 상이한 RD모델 파라메터를 갱신하게 되어, 결과적으로 Q값이 잘못 결정 되는 경우가 종종 발생한다. 이때 과도한 전송비트를 엔코딩 과정에서 발생시키게 되는 경우가 발생하며, 만약 발생된 비트량이 송신 버퍼의 제한된 용량을 초과하게 되는 경우에는, 송신 버퍼의 점유율을 낮추기 위하여 다음에 전송하려는 영상의 코딩을 중단해야 하는 문제를 야기시킨다. RD모델 갱신이 완료되면 Step 2로 돌아가서 새로운 영상 프레임에 대한 제어 과정을 반복한다.

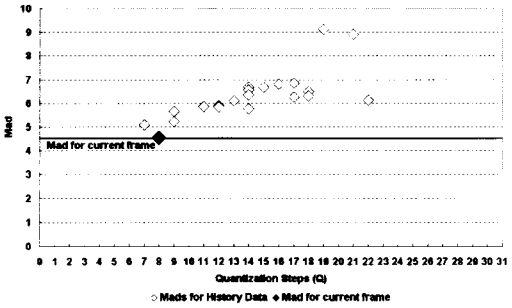
MPEG-4 프레임 기반 비트율 제어 방법에 의해 QCIF급 "Forman" 영상 시퀀스를 128kbps, 30Hz로 엔코딩을 수행한 실험 결과의 예를 그림 4에 도시하였는데, 232번째 영상 프레임의 코딩 결과를 도식화 한 것이다. 그림 4(a)에 도시된 프레임 212부터 231 프레임까지 코딩 완료되어 저장된 20개(N)의 1차원 히스토리 큐의 데이터 {Mad, Q, $Encoded\ Bits$}는 그림 4(b)와 그림 4(c)에 "○"와 "◇"로 각각 표시되어 있다. 그림 4(b)에는 또한, 히스토리 큐에 저장된 데이터("○")를 이용하여 RD모델 파라메터 X_1과 X_2를 갱신한 후 해당 파라메터 값과 현재 프레임의 영상 복잡도 Mad(=4.546835)를 식(1)에 적용하고 Q값을 1부터 31까지 변경하면서 발생된 데이터를 곡선으로 표시한 RD 모델 커브를 실선으로 도시하였으며, 코딩하려 하는 232번째 프레임의 타겟 비트량 5,223비트를 이미 갱신된 RD 모델 커브에 적용하여 8로 구해진 Q값, 그리고 해당 Q값을 이용하여 실제로 엔코딩한 후 발생된 비트량($Encoded\ Bits$) 2,416비트를 "●"로 표시하여 도시하였다. 그림 4(b)에서 쉽게 알 수 있듯이, 20개의 데이터를 적용하여 RD 모델은 적절하게 구성하였으나, 이를 232번째 프레임의 엔코딩에 적용하였을 때 발생된 비트의 양은 RD모델의 타겟 비트량과 많은 차이가 있음을 발견할 수 있다. 이는 그림 4(c)를 관찰하면 알 수 있는데, 1차원 히스토리 큐에 저장된 과거의 코딩 결과의 영상 복잡도 Mad ("◇")가 현재 코딩하려는 영상의 복잡도("◆")와 상이하게 달라서, 이미 갱신된 RD 모델의 특성과 상이한 외삽이 수행되었기 때문이다. 이는 코딩하려는 232번째 프레임의 Mad(=4.546835)가, 이미 저장된 20개의 Mad 데이터(5.080607 ≤ Mad ≤ 9.122238) 구간 내부에 위치한 보간의 경우가 아니라 외각에 위치하고 있으므로 외삽이 발생된 경우로서, 외삽의 예측 정확도는 보간시의 예측 정확도에 비해 현저히 떨어지기 때문에 발생된다.

Frame #	Mad	Q	Encoded Bits	Frame #	Mad	Q	Encoded Bits
212	6.274266	17	3192	222	6.312737	19	1592
213	6.120384	22	2480	223	6.359809	14	2736
214	8.919389	21	2640	224	6.111151	13	3152
215	9.122236	19	3160	225	5.910669	12	2912
216	6.84596	17	3224	226	5.777778	14	1944
217	6.811671	16	3152	227	5.827454	12	2960
218	6.67803	15	3360	228	5.848761	11	4000
219	6.684975	14	3400	229	5.650687	9	6072
220	6.489031	18	1872	230	5.237098	9	4216
221	6.570273	14	3160	231	5.089607	7	8080
[illegible]	[illegible]	[illegible]	[illegible]	232	4.848696	8	2416

(a) 1차원 히스토리 큐에 저장된 과거의 코딩 정보와 현재 프레임의 영상 정보



(b) 실제 발생 비트량과 양자화 스텝과의 관계로 도시한 엔코딩 결과



(c) 영상 복잡도 *Mad*와 양자화 스텝 *Q*의 관계로 도시한 엔코딩 결과

그림 4 외삽이 발생된 경우의 프레임 기반 비트 생성율 제어 결과의 예

3. 2차원 히스토리 풀과 슬라이딩 윈도우를 이용한 프레임 기반 비트율 제어 방법

엔코딩 하려는 영상 프레임의 특성이 코딩 완료되어 1차원 히스토리에 저장된 결과와 현저히 다르다면, RD 모델 갱신 후 양자화 스텝 결정 단계에서 외삽이 수행되고, 이는 보간 수행시의 결과와 비교하여 정확도가 현저히 떨어짐을 알 수 있었다. 그러므로, 가급적 외삽을 수행하는 경우를 배제해야 하는데 또는 보간을 수행하는 경우로 전환하여야 하는데, 이는 과거에 코딩 완료된 결과를 가급적 많이 저장한 후, 코딩하고자 하는 영상 프레임의 특성과 비슷한 데이터만을 이용하여 RD모델을 갱신함으로써, 외삽의 수행을 보간의 수행으로 전환할 수 있게 된다.

본 논문에서는, 2차원 히스토리 풀과 슬라이딩 윈도우를 이용한 프레임 기반 비트율 제어 방법을 제안한다. 이는, 기존의 방법이 RD 모델 갱신에 순차적 1차원 히스토리 큐만을 이용하여 영상의 특성을 적극적으로 반영하지 못한 점을 보완하여 N개의 크기를 갖는 M개의 1 차원 히스토리 큐($N * M$: 2차원 히스토리 풀)를 두어 코딩 결과를 나누어 저장하고, 코딩하려는 영상프레임의 특성에 적합한 정보들로 히스토리를 재구성하고, 이를 회귀 과정에 이용하여 RD 모델을 적절하게 갱신하는 비트율 제어 방법이다.

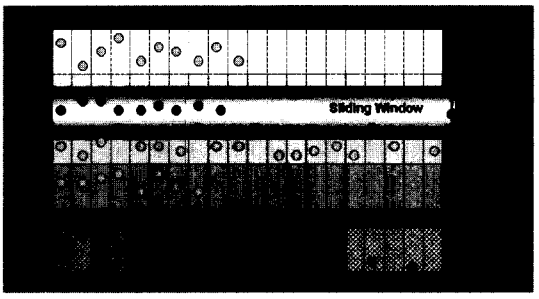


그림 5 2차원 히스토리 풀과 워킹 히스토리 큐

2차원 히스토리 풀의 구조와 워킹 히스토리 큐를 그림 5에 도시하고 있다. 그림 5를 통해서 알 수 있듯이, 2차원 히스토리 풀은 N개의 크기를 갖는 1차원 히스토리 큐를 M개 포함하여 구성된다. 코딩이 완료된 영상 데이터들은 프레임 별로 {*Mad, Q, Encoded Bits*}의 3개의 코딩 정보로 구성되어있으며 각각의 코딩 결과는 해당 영상 복잡도 *Mad*의 크기에 따라 지정된 FCFS 히스토리 큐에 분산되어 저장된다. 이는 시간에 따라 순차적으로 히스토리 큐에 코딩 결과가 저장되는 기존의 방법과 달리, 2차원 히스토리 풀에 영상 복잡도에 따라 분산되어 저장되는 방법으로서 동일 히스토리 큐에 저장될 때는 시간적 순서에 의해 순차적으로 저장된다. 그림 5에서 보는 바와 같이 첫번째 히스토리 큐에는 $L_0 \leq Mad < L_1$구간의 코딩 결과가 순차적으로 저장되며, 두번째 히스토리 큐에는 $L_1 \leq Mad < L_2$의 코딩 결과가, M번째 히스토리 큐에는 $L_{M-1} \leq Mad < L_M$구간의 코

딩 결과가 저장된다. 만약 특정 히스토리 큐에 저장된 정보의 수가 N을 초과할 때에는 FCFS 큐로 구성되어 있으므로 가장 오래 저장된 정보를 삭제한다.

코딩 하고자 하는 영상 프레임의 특성과 비슷한 데이터만을 이용하여 RD 모델을 갱신하기 위해서는 2차원 히스토리 풀에 저장된 정보 중 현재 프레임의 영상 복잡도 *Mad*값과 유사한 특징을 갖는 코딩 결과 {*Mad, Q, Encoded Bits*}를 선택하여 히스토리를 재구성할 필요가 있다. 이때 슬라이딩 윈도우를 사용하게 되는데, 현재 코딩하려는 프레임의 *Mad*값에 슬라이딩 윈도우의 중앙부를 일치시키고, RD모델 갱신에 필요한 정보를 수집하여 워킹 히스토리 큐(Working History Queue)에 저장하게 된다. 이때 슬라이딩 윈도우의 폭은 현재 프레임의 *Mad*값 $\pm\ \alpha$이며 최대로 선택 가능한 코딩 결과의 수는 워킹 히스토리 큐의 크기인 N개까지 가능하다. 워킹 히스토리 큐에 재 구성된 코딩 결과를 이용하여 현재 프레임을 엔코딩하기 위한 RD모델을 회귀 과정을 이용하여 적절하게 갱신하는 비트율 제어 방법이다.

양자화 스텝 예측을 위한 워킹 히스토리는 다음과 같은 5 가지의 구성 단계를 거친다.

1) 현재 엔코딩하려는 프레임에 대한 *Mad*와 2차원 히스토리 풀의 *Mad*와의 차가 $\pm\alpha$(=3) 이내의 *Mad*값을 갖는 정보들을 2차원 히스토리 풀로부터 선택하며, 최대 개수는 워킹 히스토리 큐의 크기인 N개까지 가능하다.

2) 첫 번째의 과정에서 구한 데이터의 수가 너무 적다면 RD모델 예측을 수행할 수 없으므로 데이터의 수가 5개 이상의 적정 수준 이상이 될 때까지 슬라이딩 윈도우의 폭을 넓혀가며, 즉 프레임에 대한 *Mad*와 2차원 히스토리 풀의 *Mad*에 대한 차의 기준을 넓혀가며 데이터를 수집한다: 이는 슬라이딩 윈도우의 폭을 α로 정하고 그 범위안의 코딩 결과를 수집할 때, 엔코딩을 수행하는 영상시퀀스의 특성이 급변하는 경우 또는 특정 *Mad*구간을 관장하는 히스토리 큐의 데이터가 부족한 경우, RD모델을 예측하는데 필요한 데이터를 충분하게 구할 수 없는 경우가 발생할 수도 있기 때문이며 이때에는 슬라이딩 윈도우의 폭을 넓혀가며, RD모델 예측에 필요한 데이터를 충분히 확보할 필요가 있다. 실제로 RD 모델 예측에는 최소 2개 이상의 데이터가 있으면 가능하지만, 보다 정확한 예측을 하기 위하여 5개로 결정하였다.

3) 워킹 히스토리에 저장된 데이터들을 엔코딩된 순서대로 정렬한다. 이는 시간상으로 현재 입력영상에 가까운 데이터들이 멀리 떨어진 데이터들에 비하여 영상 특성이 비슷하기 때문이다.

4) 워킹 히스토리에 저장된 데이터들의 양자화 스텝이 한 값으로 편중되는 경우가 발생되면, 슬라이딩 윈도우의 폭을 확대하여 편중된 데이터 중 현재 입력영상과의 시간상으로 제일 오래된 k개(5개)를 선택하여 교체한다. 이는 데이터들의 특정 양자화 스텝에 대한 편중화를 방지하기 위함이다.

5) 가장 최근에 엔코딩된 두개의 데이터를 이용하여, 최근에 엔코딩된 영상 특성과 유사하지 않는 정보들을 제거한다. 이때 *RATIO* 함수를 이용하는데, 다음은 *RATIO* 함수의 이용 과정을 나타내고 있다.

Mad_i : i번째 영상 프레임의 영상 복잡도(*Mad*)
Bit_i : i번째 영상 프레임의 타겟비트(*Bits*)
Q_i : i번째 영상 프레임의 양자화 스텝 (*Q*)
Last: 코딩하고자 하는 영상 프레임의 바로 전 프레임에 대한 인덱스

(a) 워킹 히스토리 내의 데이터들에 대하여 영상 특성 비율(*RATIO*)을 구한다.
$RATIO_i = Bit_i * Q_i / Mad_i$
(b) 가장 최근에 코딩된 두 개의 영상 특성 비율(*RATIO*) 값의 평균($RATIO_{AVE}$)을 구한다.
$RATIO_{AVE} = (RATIO_{Last} + RATIO_{Last-1}) / 2$
(c) 상기 비율(*RATIO*)에 대한 편차(*Sigma*)를 구한다. n은 워킹 히스토리 내의 데이터 수이다. 함수 *sqrt* 는 제곱근을 구한다.
for $(i = 1,\ i \le n;\ i{+}{+})$ $SUM = SUM + (RATIO_i - RATIO_{AVE})$
$Sigma = Sqrt(SUM^2 / n)$
(d) 만일, $ABS(RATIO_i - RATIO_{AVE})$ 율이 상기 *Sigma* 보다 작으면 (최근에 코딩된 영상 특성 비율이 비슷하면), 워킹 히스토리에 잔존 시키고 그렇지 않으면 제거한다. *ABS(x)*는 x에 대한 절대값이다.

본 논문에서 제안된 프레임 기반 비트율 제어 과정을 그림 6에 블록도로 도시하였다. 기존의 MPEG-4 프레임 기반 비트율 제어 방법은 현재 프레임을 코딩하기 전에 RD 모델을 갱신하며 해당 모델을 현재 프레임의 엔코딩에 적용하는 사후갱신(Post-Update)방법이며, 본 논문에서 제안하는 방법은 RD모델 갱신과정을 현재 프레임의 영상 복잡도와 유사한 데이터를 2차원 히스토리 풀로부터 선택하여 그들을 이용하여 RD모델을 갱신한 후 현재 프레임에 대한 엔코딩을 수행하는 사전갱신(Pre-Update)방법으로 엔코딩 이후에는 2차원 히스토리 풀의 데이터 갱신만을 수행하는 방법이다.

제안된 비트율 제어 방법은 7단계로 다음과 같이 구성되어 있다.

Step 1. 초기화

RD 모델 파라메터 X_1과 X_2를 초기화 한다.

Step 2. 프레임에 대한 타겟 비트량 계산

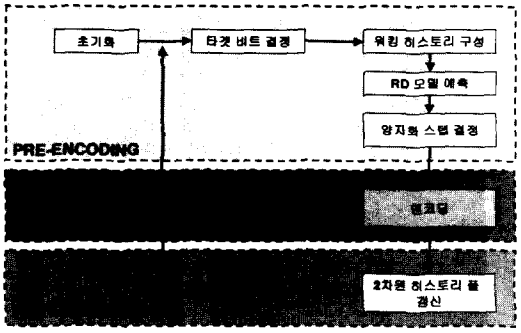


그림 6 제안된 프레임 기반 비트율 제어 방법

2장에서 설명된 MPEG-4 VM 프레임 기반 비트율 제어 방법의 Step 2와 동일하다.

Step 3. 워킹 히스토리 구성

엔코딩하려는 프레임의 *Mad*를 기준으로 오차를 구하여 그 차가 $\pm\alpha$ 이내의 *Mad*값을 갖는 코딩 정보를 슬라이딩 윈도우를 이용하여 2차원 히스토리 풀로부터 선택한 후 워킹 히스토리에 저장한다.

Step 4. RD 모델 예측

워킹 히스토리 큐를 기반으로 회귀 과정을 수행하여 RD 모델을 예측한다. 이때 사용된 회귀방법은 MPEG-4 프레임 기반 제어 방법과 동일하게 수행한다.

Step 5. 양자화 스텝 결정

2장에서 설명된 MPEG-4 프레임 기반 제어 방법의 Step 3와 동일하다. 단 이전 영상과 현재 영상의 복잡도 *Mad*가 β (=3) 이상 차이가 날 경우는 영상의 특성이 급격히 변화하는 경우로서 *Q*값의 변화를 ±25%의 범위로 제약하는 규칙을 적용 하지 않는다.

Step 6. 프레임에 대한 엔코딩

프레임단위의 엔코딩을 수행한다.

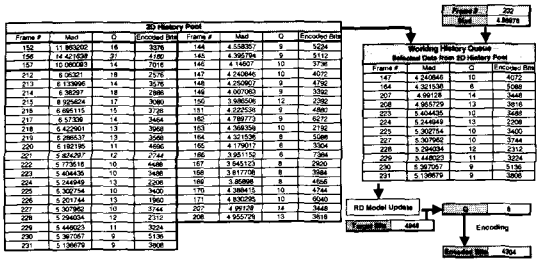
Step 7. 히스토리 풀 갱신

엔코딩 결과(*Mad, Bits, Q*)를 *Mad*를 기준으로 2차원 히스토리 풀에 저장한다.

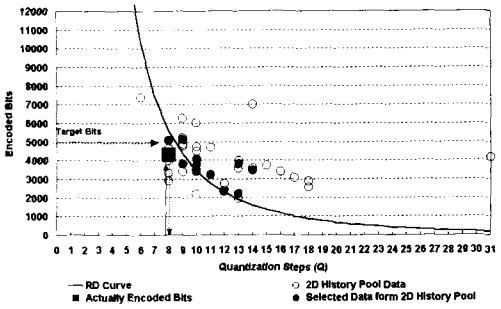
만약 현재 엔코더 단의 버퍼의 80% 이상을 사용한 상태에 이르면 오버플로우를 방지하기 위해 다음 영상 프레임의 엔코딩을 스킵한다.

제안된 제어 방법을 구현하기 위해서는, 기존의 MPEG-4프레임 기반 제어 방법에 비하여$N*M$의 2차원 히스토리 풀 구성에 필요한 메모리가 추가로 요구된다. 예를들어 *Q*값에 1 byte (5 bit소요), *Mad*에 4 byte (float), *Bit*에 2 byte(integer)가 필요하므로 약 $(1+4+2)*N*M$ byte가량의 메모리가 추가로 소요된다. 제안된 방법을 사용하여 그림 4에서 사용된 QCIF급 "Forman" 영상 시퀀스를 128kbps, 30Hz로 엔코딩을 수행한 실험 결과의 예를 그림 7에 도시하였는데, 동일한 232번째 영상 프레임에 대한 코딩 결과를 도식화 한 것이다. 제안된 방법의 2차원 히스토리 풀은 *Mad*의 크기에 따라 20개의 공간을 가진 3개의 1차원 히스토리 풀($N*M = 20*3$)로 구성되어있으며, *Mad*구간은 0~5, 5~10,그리고 10이상의 3개의 구간으로 나누었다. 그림 7(a)에서는, 이전 영상의 코딩 결과(*Mad, Q, Encoded Bits*)를 저장하고 있는 2차원 히스토리 풀로부터, 현재 엔코딩을 수행하려는 232번째 영상 프레임의 *Mad* (4.98973) 값과 유사한 영상 복잡도를 갖는 코딩 결과를 슬라이딩 윈도우를 사용하여 추출하여 워킹 히스토리 큐를 재구성하는 과정을 도시하였다. 현재 영상 프레임의 *Mad*값은 그림 7(c)에 "■"로 표시되어있다. 2차원 히스토리 풀을 살펴보면, $0 \le Mad < 5$인 코딩 결과가 20개, $5 \le Mad < 10$인 코딩 결과가 20개, $Mad \ge 10$인 코딩 결과가 3개 저장되어 있음을 알 수 있으며 워킹 히스토리 큐는 매 영상 프레임에 대한 엔코딩을 수행할 때마다 갱신된다. 그림 7(b)와 (c)에는 2차원 히스토리 풀에 저장된 데이터가 "○"로 표시되어 있으며, 슬라이딩 윈도우에 의해 선택된 워킹 히스토리 큐의 데이터는 "●"로 표시되어 있다. 워킹 히스토리 큐에 저장된 데이터("●")를 이용하여 RD모델 파라메터를 갱신한 후 결정된 RD 모델 커브를 그림 7(b)에 실선으로 도시하였다. 코딩하려는 232번째 프레임의 타겟 비트량(4,948 비트)을 갱신된 RD모델에 적용하여 *Q*값을 8로 결정하였으며, 해당 *Q*값을 이용하여 실제 엔코딩이 완료된 후에 발생된 비트량(4,304비트)을 그림 7(b)에 "■"로 도시하였다. 그림 7(b)에서 알 수 있는 바와 같이 타겟 비트량과 유사한 발생 비트량을 얻어 낼 수 있었는데 이는 현재 프레임의 영상 복잡도와 유사한 특징을 갖는 과거의 코딩 데이터를 선별하여 RD 모델을 갱신하므로서 이루어 질 수 있으며, 그림 7(c)에 도시된 바와 같이 현재의 영상 특성("■": *Mad* = 4.98973)이 워킹 히스토리 큐에 저장된 데이터("●")의 영상 특성 구간 내 ($4.240846 \le Mad \le 5.446023$)에 위치한 보간의 경우이므로 보다 정확한 예측이 가능해졌기 때문이다.

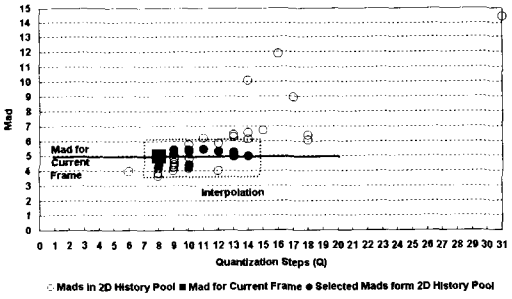
RD모델 갱신시 실제 발생된 보간과 외삽의 또 다른 실험 예를 그림 8의 (a)~(d)에 도시하였다. 그림 8의 (a)와 (c)는 각각 SIF급 "Stefan" 영상 시퀀스를 1,024kbps, 30Hz로 엔코딩을 수행할 때 235번째 영상 프레임의 코딩시 실제 발생된 외삽 및 보간이 수행된 경우의 예로서, 결정된 *Q*값과 실제 발생 비트량의 관계를 나타내는 그래프들이며, (b)와 (d)는 *Mad*와 *Q*값과



(a) 2차원 히스토리 풀에 저장된 코딩 정보로부터 워킹 히스토리 구성 및 엔코딩 과정



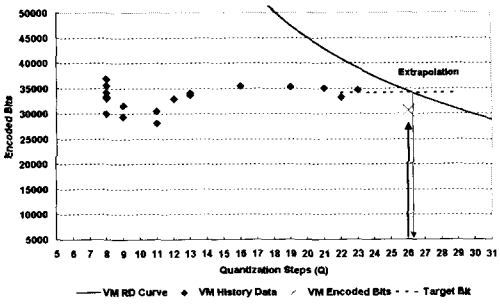
(b) 실제 발생 비트량과 양자화 스텝과의 관계로 도시한 엔코딩 결과



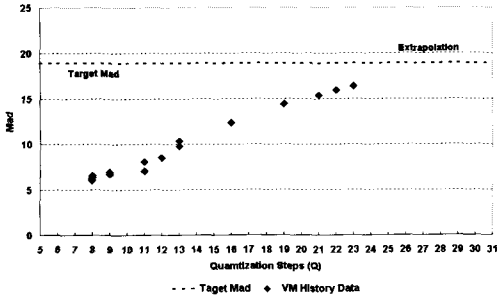
(c) 영상 복잡도 *Mad*와 양자화 스텝 Q의 관계로 도시한 엔코딩 결과

그림 7 보간이 발생되도록 제안된 프레임 기반 비트 생성율 제어 결과의 예

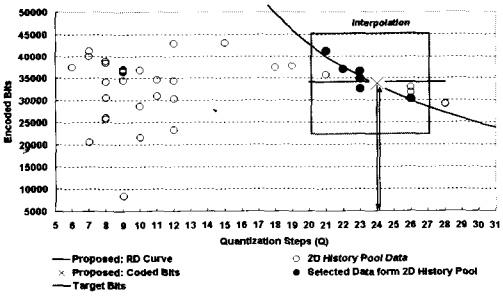
의 관계로 살펴본 외삽 및 보간을 나타내는 그래프들이다. 그림 8의 (a)와 (b)는 기존의 방법을 비트율 제어 방법을 이용하여 235번째 프레임에 대한 엔코딩을 수행할 때 외삽이 발생된 경우이다. 그림 8(a)와 (b)의 "◆"는 1차원 히스토리 풀에 저장된 데이터로 그림 8(a)에서 곡선으로 표시된 RD모델 갱신시 사용되었으며, 해당 모델을 이용하여 Q값을 26으로 결정하였고 실제 "×"로 표시된 약 30,000비트의 비트량이 발생된 경우를 도시하



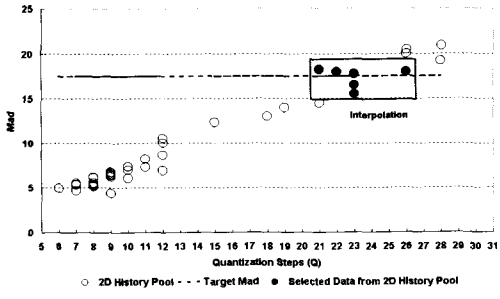
(a) 외삽시의 Q 스텝 예측



(b) *Mad*로 도시한 외삽



(c) 보간시의 Q 스텝 예측



(d) *Mad*로 도시한 보간

그림 8 "Stefan"영상 시퀀스의 235번째 프레임에 대한 엔코딩 수행 과정에서 RD 모델 갱신시 실제로 발생된 외삽과 보간의 예.

고 있다. 이는 해당 프레임의 타겟 비트량이 35,000비트로서 약 5,000비트의 예측 오차를 보여주는 경우이다. 그림 8의 (c)와 (d)는 본 논문에서 제안된 방법을 이용하여 동일한 영상 프레임을 엔코딩 할 때 보간이 발생된 경우를 도시한 것으로서, "○"는 2차원 히스토리 풀에 저장된 데이터를 나타내고 "●"는 워킹 히스토리 큐에 저장된 RD모델 갱신을 위하여 선택된 데이터를 표시한다. 갱신된 RD 모델은 그림 8(c)에서 곡선으로 표기하였다. RD모델을 이용하여 엔코딩하려는 프레임의 타겟비트(약 35,000비트)에 적합한 Q값을 24로 선택한 후 해당 Q값을 이용하여 실제 엔코딩을 수행한 결과를 "×"로 표기하였는데 타겟 비트와 거의 동일한 발생 비트를 얻어낼 수 있음을 보여주고 있다.

4. 실험 결과

제안된 비트율 제어 방법은 MOMUSYS의 MPEG-4 CODEC에 설치되었으며, 제어 성능은 MPEG-4 검증모델 (Verification Model: VM)에서 권장하는 프레임 기반 비트율 제어 방법의 성능과 비교하였다.

실험에 사용된 영상 시퀀스는 MPEG 성능평가 테스트에 광범위하게 사용되는 "News", "Silent", "Akiyo", "Weather", "Hall Monitor", "Foreman", "Stefan" 그리고 "Container"의 QCIF급 이미지 시퀀스를 이용하여 실험을 수행하였다. 사용된 시퀀스는 초당 30Hz의 프레임율 (frame/second)을 갖고 있으며, 10초 분량으로 구성되어 있다. 영상에 대한 타겟 비트율 할당과, 프레임 스킵 조건, 그리고 16으로 지정한 움직임 벡터의 범위 등은 동일하게 조정 하였다. 첫번째 프레임은 Q를 10으로 고정하여 인트라코딩(Intra Coding)을 수행하였고, 두번째 프레임은 Q를 10으로 고정한 인터코딩(Inter Coding)을 수행하였으며, 나머지 프레임은 모두 프레임 기반 비트 생성율 제어 방법을 적용한 인터코딩을 수행하여 결과를 도출하였다.

제안된 방법의 2차원 히스토리 풀은 QCIF영상에 적합하게 Mad의 크기에 따라 0~3, 3~6, 6~9, 9~12, 그리고 12이상의 5구역으로 나누었으며 (M=5), N은 20으로 고정하였다.

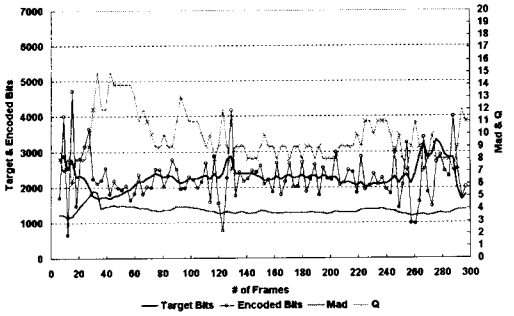
제안된 비트율 제어 방법과 MPEG-4 VM 비트율 제어 방법의 실험 결과 비교를 표 1에 도시하였다. 결과는 프레임 당 생성된 평균 비트량 (Ave. Bits/Frame), Y(luminance)신호의 평균 피크 신호 대 잡음비 (Peak Signal to Noise Ratio: PSNR in dB), 10초당 프레임 스킵의 횟수를 10Hz에서 24kbps, 30Hz에서 48kbps, 그리고 30Hz에서 64kbps등의 전송 조건을 다양하게 변화시키면서 비교실험을 수행 하였다. 예로서 시퀀스가 10 Hz로 엔코딩되면서 초당 24,000bit로 전송되는 경우에는 만약 스킵이 발생되지않고 모든 영상 프레임이 전송되었다면 평균 Bits/Frame은 2400bit이다.

표에서 제안된 방법의 성능이 Bits/Frame비교에서 기존의 프레임 기반 제어 방법보다 정확하며, 비슷한 PSNR을 유지하면서도 매우 적은 프레임 스킵을 발생시키는 것을 쉽게 발견할 수 있다. 이는 제안된 방법의 영상에 대한 양자화 예측이 기존의 방법에 비하여 적합하게 수행되었음을 의미한다.

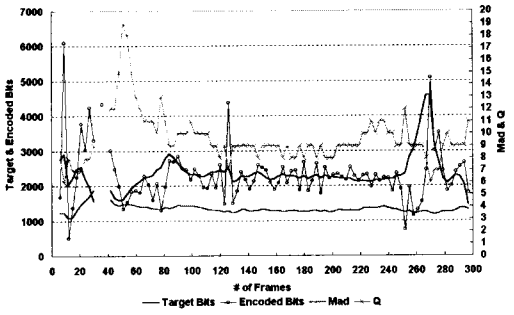
표 1 프레임 기반에서의 MPEG-4 VM 비트율 제어와 제안된 비트율 제어 방법의 성능 비교

Bitrates Hz	Sequence	Methods	Ave. Bits/ Frame	PSNR (Y)	Frame Skips
24kbps 10Hz 2400 Bits/Frame (average)	Silent	MPEG-4 VM	2524.21	30.93	5
		Proposed	2396.32	30.64	0
	Akiyo	MPEG-4 VM	2564.68	38.39	6
		Proposed	2438.71	38.48	1
	Hall Monitor	MPEG-4 VM	2451.92	34.32	2
		Proposed	2401.36	34.28	0
48kbps 30Hz 1600 Bits/Frame (average)	News	MPEG-4 VM	1655.53	32.37	10
		Proposed	1617.37	32.47	2
	Weather	MPEG-4 VM	1654.85	37.06	7
		Proposed	1628.42	37.17	3
	Container	MPEG-4 VM	1791.05	34.46	31
		Proposed	1659.18	34.48	8
64kbps 30Hz 2133 Bits/Frame	News	MPEG-4 VM	2224.83	33.61	12
		Proposed	2139.23	33.74	0
	Silent	MPEG-4 VM	2158.11	33.20	4
		Proposed	2131.60	33.18	0

표 1의 실험 결과 중에 "Hall monitor" QCIF 시퀀스에 대해 24kbps 10Hz 전송 환경에서 비트율 제어 결과 비교를 그림 9에 도시하였는데, 코딩 결과 {Mad, $Target\ Bits$, $Encoded\ Bits$, Q}를 엔코딩된 프레임 순서에 따라서 도시하였다. 그림 9의 (a)에는 제안된 방법에 의한 엔코딩 결과를 나타내었고, (b)에는 MPEG-4 프레임 기반 비트율 제어 방법에 의한 엔코딩 결과를 나타내었다. 그림 9에서, 제안된 방법의 양자화 예측이 기존의 방법에 비하여 적합하게 수행되고 있음을 쉽게 발견할 수 있다.



(a) 제안된 방법의 {*Mad, Target Bits, Encoded Bits, Q*} 그래프



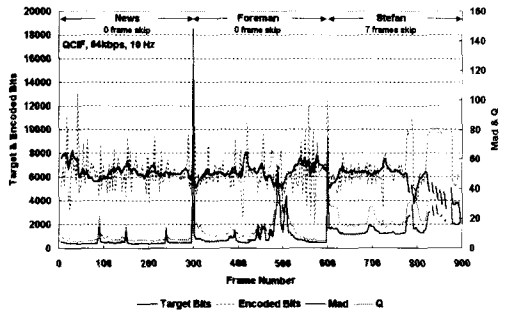
(b) VM 비트율 제어에서의 {*Mad, Target Bits, Encoded Bits, Q*} 그래프

그림 9 "Hall monitor" QCIF 시퀀스에 대해 24kbps 10Hz 전송환경에서 제안된 방법의 비트율 제어와 VM 비트율 제어 결과 비교

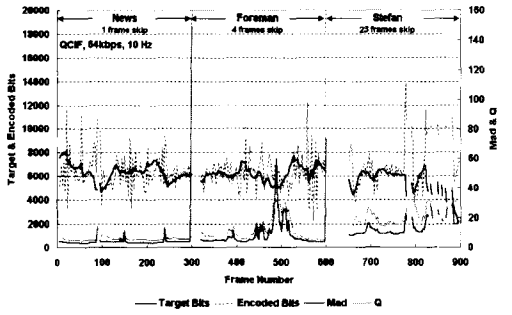
실험에 사용된 영상 시퀀스 중에서 영상 특성 변화가 급격하게 일어나는 "News", "Foreman" 그리고 "Stefan" QCIF 시퀀스를 조합하여 하나의 "News + Foreman + Stefan" QCIF 시퀀스로 만들고, 64kbps, 10Hz의 전송환경에서 해당 시퀀스의 비트율 제어 결과 비교를 그림 10에 도시하였다. 사용된 시퀀스는 초당 30Hz의 프레임율을 갖고 있으며, 3개의 시퀀스를 조합하였으므로 30초 분량으로 구성되어 있다. 조합된 시퀀스의 영상 특성 변화는 그림 10(a)와 (b)에서 살펴볼 수 있는데 영상 복잡도 *Mad*의 변화추이를 보면 "News" 시퀀스에서 $2.8 \leq Mad \leq 14.1$, "Foreman" 시퀀스에서 $3.5 \leq Mad \leq 60$, 그리고 "Stefan" 시퀀스에서는 $8.3 \leq Mad \leq 37$까지 영상 특성이 급격하게 변화하는 시퀀스 3개가 조합되어 있음을 살펴볼 수 있다. 또한 "News" 시퀀스에서 "Foreman" 시퀀스로의 장면전환이 발생되었을 때의 *Mad*값은 약 150까지 갑자기 증가되며, "Foreman" 시퀀스에서 "Stefan" 시퀀스로의 장면전환이 발생되었을 때의 *Mad*값은 약 74로 증가함을 살펴볼 수 있다. 즉 실험을 위하여 조합된 "News + Foreman + Stefan" 영상 시퀀스는, 갑작스러운 장면전환에 의한 영상 특성의 급격한 변화로 외삽의 경우가 발생되는 경우뿐만 아니라, 엔코딩을 하려는 영상의 특성이 FCFS 큐에 저장된 엔코딩 결과와 매우 상이한 경우, 또는 비슷한 엔코딩 결과가 과거에 존재하였지만 FCFS 큐의 크기가 제한되어 해당 프레임의 영상 특성과 비슷한 결과를 저장하지 못하고 있는 경우에 외삽의 경우가 종종 발생하는 경우를 포괄적으로 실험할 수 있다.

비교실험은 앞선 실험과 동일한 조건으로 수행하였으며, 첫 번째 프레임은 Q를 10으로 고정하여 인트라코딩을 수행하였고, 두 번째 프레임은 Q를 10으로 고정한 인터코딩을 수행하였으며, 나머지는 모두 프레임 기반 비트 생성율 제어 방법을 적용한 인터코딩을 수행하여 비교 결과를 도출하였다. 제안된 2차원 히스토리 풀의 구성도 앞선 실험과 동일하게 *Mad*의 크기에 따라 0~3, 3~6, 6~9, 9~12,그리고 12이상의 5구역으로 나누었으며, N은 20으로 고정하였다.

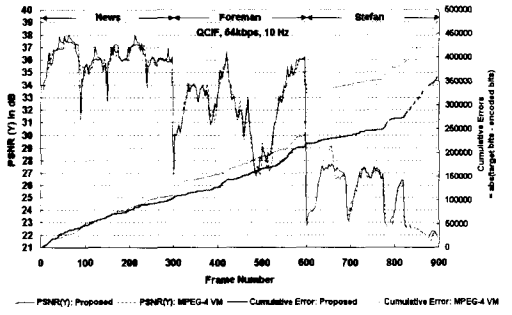
그림 10(a)와 (b)에는 각각 제안된 방법의 엔코딩 결과와 MPEG-4 프레임 기반 비트율 제어 방법에 의한 엔코딩 결과를 도시하였으며, 엔코딩 프레임 순서에 따라 도출된 {*Mad, Target Bits, Encoded Bits, Q*}의 그래프를 도시하였다. 그림 10에서 쉽게 발견할 수 있는 바와 같이 제안된 방법은 "Stefan" 시퀀스에서만 총 7개의 프레임 스킵이 유발된 반면, 비교된 MPEG-4 VM 방법은 "News" 시퀀스에서 1개, "Foreman" 시퀀스로의 장면전환 직후에 4개, 또한 "Stefan" 시퀀스로 장면전환 후 23개 등 총 28개의 영상 프레임이 화면 출력(display)이 되지 않고 스킵되는 비트율 제어를 수행한 결과를 보여주고 있다. 또한 제안된 방법이 제어하지 못한 "Stefan" 시퀀스의 끝부분에서 발생된 6개의 프레임 스킵의 경우는 그림 10(a)를 살펴보면 알 수 있듯이 Q 값을 31로 연속하여 엔코딩을 수행하여도 64kbps로는 비트 생성율 제어가 가능하지 않는 급작스러운 영상변화에 기인하며 이는 MPEG-4 VM의 경우에도 동일하게 해당된다. 즉 어쩔 수 없는 6개의 프레임 스킵을 제외하면 제안된 방법은 총 1개, MPEG-4 VM은 총 22개의 프레임 스킵이 유발되었음을 알 수 있다. 그림 10(c)에서는 제안된 방법과 비교된 방법의 비트율 제어 결과로 도출된 재생 영상의 PSNR 비교와 타겟 비트와 실제 발생된 비트와의 절대차를 구하여 그 누적오차 (Cumulative Errors = abs(Target Bits - Encoded



(a) 제안된 방법의 {*Mad, Target Bits, Encoded Bits, Q*} 그래프



(b) VM 비트율 제어에서의 {*Mad, Target Bits, Encoded Bits, Q*} 그래프



(c) 제안된 방법과 VM비트율 제어 결과로 도출된 PSNR과 누적 오차 비교

그림 10 "News + Foreman + Stefan" QCIF 시퀀스에 대해 64kbps 10Hz 전송 환경에서 제안된 방법의 비트율 제어와 VM 비트율 제어 결과 비교

Bits))를 비교한 그래프를 도시하고 있다. 단 PSNR 비교와 누적 오차 비교에서 스킵이 발생된 프레임의결과는 비교하지 않았다. PSNR비교에 있어서는 제안된 방법과 MPEG-4 VM의 결과가 프레임 스킵이 발생된 프레임을 제외하게 되면 거의 유사한 것으로 보여주고 있다. 하지만 누적오차 비교에서는 제안된 방법의 누적 오차가 357,987 비트이며 MPEG-4 VM은 476,988비트로서, 제안된 방법의 양자화 예측이 기존의 방법에 비하여 적합하게 수행되고 있음을 쉽게 발견할 수 있다.

"News + Foreman + Stefan" QCIF 시퀀스를 64kbps, 10Hz의 전송환경에서 MPEG-4 프레임 기반 제어 방법을 이용하여 엔코딩을 수행할 때 FCFS 큐의 크기를 60개(N=60)로 확장하여 그림 10(b)와 같은 실험결과를 다시 한번 도출하여 보았다. 이는 제안된 방법의 2차원 히스토리 풀의 크기가 비교된 방법의 FCFS 큐의 크기보다 상대적으로 크기때문에 비교된 방법의 큐의 크기도 확장해 본 것이다. 60개의 크기로 확장된 FCFS 큐를 이용한 제어 결과는 그림 10(b)와 거의 동일하며 단지 "News" 시퀀스에서 프레임 스킵은 발생되지 않았지만, "Foreman" 시퀀스로의 장면 전환 후 5개의 프레임 스킵이, "Stefan" 시퀀스로 장면 전환 후 24개의 프레임 스킵이 유발되어 20개의 크기를 가진 FCFS큐를 이용한 제어 결과보다 1개의 프레임 스킵이 더 발생되었다. 이는 단순히 FCFS 큐의 크기를 확장하는 것은 외삽이 수행되는 경우의 수를 줄여주지 못하여 영상 특성 변화에 효율적으로 대처하지 못한다는 것을 의미한다.

5. 결 론

본 논문에서 제안한 프레임 기반 비트율 제어 방법은 코딩 완료된 과거의 영상 특성 정보를 영상 복잡도에 따라 분류하여 2차원 히스토리 풀에 저장한 후, 이를 코딩 하고자 하는 영상 특성에 적합하게 재 구성한 후 RD모델 갱신을 수행하여 엔코딩을 수행하는 비트율 제어 방법이다. 제안된 방법은 기존의 비트율 제어 방법에 비하여 영상 특성의 변화에 효과적으로 대처하여 외삽이 수행되는 것을 줄여줌으로써 정확한 양자화 스텝 결정을 수행할 수 있도록 유도하여 상대적으로 우수한 코딩 결과를 갖게 한다. 제안된 비트 생성율 제어 방법과 기존의 MPEG-4 프레임 기반 비트 생성율 제어 방법의 성능 비교에 있어서 비슷한 피크 신호대 잡음비를 유지하면서 제안된 방법의 경우 프레임 당 발생된 비트량과 타겟 비트량의 오차가 0.2%~3.7%인 반면, 비교된 방법의 경우는 1.2%~11.9%로서 제안된 방법이 보다 정확하게 프레임 당 비트 생성율 제어를 수행하고 있음을 알 수 있었다. 또한 제안된 방법이 상대적으로 프레임 스킵을 약 2배에서 12배 정도 줄여줄 수 있었다

참 고 문 헌

[1] W. Ding, B. Liu, "Joint encoder and channel rate control of VBR video over ATM networks," *Proceedings of the Society of Photo-optical Instrumentation Engineers*, issue 2668, pp. 392~407, 1996.

[2] W. Li, J.-R. Ohm, M. v Schaar, H. Jiang, S. Li, "MPEG-4 Video Verification Model version 18.0," *ISO/IEC/ JTC1/SC29/WG11 N3908*, 2001

[3] T. Chiang and Y.-Q. Zhang, "A New Rate Control Scheme Using Quadratic Rate Distortion Model," *IEEE Transactions on circuits and systems for video technology*, vol. 7, no. 1, pp. 246~250, 1997

[4] S.-G. Ryoo, S.-J. Kim, Y.-S. Seo, "Rate Control Tool: Based on Human Visual Sensitivity(HVS) for Low Bitrate Coding," *ISO/IEC/JTC1/SC29/WG11 MPEG96/0566*, 1996

[5] Y. M. Chien, "Suboptimal quantization control emplying approximate distortion-rate relations for motion video coding," *SPIE* vol. 3024, pp. 138~148, 1997

[6] L. Wang, "Rate control for MPEG video coding," *Signal Processing, Image Communication*, vol. 15, no. 6, pp 493~511, 2000.

[7] T. Y. Kim, B. H. Roh, J. K. Kim, "An accurate bit-rate control for real-time MPEG video encoder," *Signal Processing, Image Communication*, vol. 15, no. 6, pp. 479~492, 2000.

[8] H. Sun, W. Kwok, M. Chien, and C. H. John, "MPEG Coding Performance Improvement by Jointly Optimizing Coding Mode Decisions and Rate Control," *IEEE Transactions on circuits and systems for video technology*, vol. 7, no. 3, pp. 449~458, 1997

[9] D. Bagni, G. A. Mian, S. Tono, "Efficient Intra-frame Encoding and improved Rate Control in H.263 compatible format," *Progress in Connectionist-Based Information Systems*, pp. 767~774, 1997.

[10] H.-J. Lee, T. Chiang and Y.-Q. Zhang, "Multiple-VO rate control and B-VO rate control," *ISO/IEC/JTC1/SC29/WG11 MPEG97/M2554*, 1997

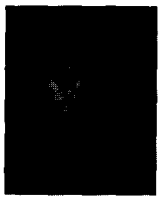
[11] J. W. Lee, "New MPEG-2 rate control algorithm based on motion estimation statistics," *SPIE* vol. 3309, pp. 395~404, 1997

[12] J. I. Ronda, M. Eckert, S. Rieke, F. Jaureguizar, A. Pacheco, "Advanced rate control for MPEG-4 Coders," *SPIE* vol. 3309, pp. 383~394, 1997

[13] J. I. Ronda, M. Eckert, F. Jaureguizar, N. Garcia, "Rate control and bit allocation for MPEG-4," *IEEE Trans. On Circuits and Systems for Video Technology*, v. 9, pp. 1243~1258, 1999

박 광 훈

1985년 연세대학교 전자공학과(공학사). 1987년 연세대학교 전자공학과(공학석사). 1991년 Case Western Reserve Univ., Dept. of EEAP (M.S.). 1995년 Case Western Reserve Univ., Dept. of EEAP (Ph.D.). 1995년 ~ 1997년 현대전자 멀티미디어 연구소 책임연구원. 1997년 ~ 2001년 연세대학교 문리대학 전산학과 부교수. 2001년 ~ 현재 경희대학교 컴퓨터공학과 부교수. 관심분야는 멀티미디어, 비디오 신호처리, 패턴인식, 영상처리, 계산지능 등



이 윤 진

1999년 연세대학교 전산학과(이학사). 2001년 연세대학교 전산학과(이학석사). 2001년 ~ 현재 경희대학교 컴퓨터공학과 박사과정. 관심분야는 멀티미디어, 비디오 신호처리, 영상처리