

비디오 코딩을 위한 적응 양자화 알고리즘

정희원 조남익*, 이상용**

Adaptive Quantization Algorithms for Video Coding

Nam Ik Cho*, Sang Yong Lee** *Regular Members*

요 약

본 논문에서는 비디오 코딩에서 앞 프레임으로부터 현재 프레임의 양자화 오차에 대한 정보를 얻어서 양자화 간격을 조절하는 적응 양자화 알고리즘을 제안한다. 비디오 인코더에서는 앞 프레임의 원 영상과 복원 영상이 프레임 메모리에 저장되어 있는 상태이므로 이들의 오차로부터 현재 프레임의 양자화 오차를 예측하는 것이다. 현재 코딩되고 있는 매크로블록(MB)의 에러를 예측하기 위해서 앞 프레임에서 이와 특성이 같은 매크로 블록을 찾고, 여기서의 원 영상과 복원 영상의 차이를 현재 MB의 에러로 간주한다. 이렇게 하여 현 프레임에서 각 MB의 에러를 구하고 현재 코딩되고 있는 MB의 에러를 전체 MB의 에러 평균과 비교해서 평균보다 크면 작은 스텝 크기가, 작은 경우에는 큰 스텝 크기가 현재의 MB에 주어진다. 그 결과 현 MB의 에러 분포가 평균에 더 집중되기 때문에 에러의 분산값이 낮아지고 영상의 질은 좋아진다. 여기서 에러는 일반적인 SAD(sum of absolute difference) 외에도 블록 현상을 줄이기 위한 에러 측정치도 고려되었다. 특히 낮은 비트율의 응용에 있어서 제안된 알고리즘은 TM5와 비교할 때 더 작은 에러 분산과 더 높은 PSNR 값을 보였다. 또한 보다 효율적인 하드웨어 구현을 위하여 수정된 알고리즘도 제시한다.

ABSTRACT

This paper proposes adaptive quantization algorithms for video coding using the information obtained from the previously encoded image. Before quantizing the DCT coefficients, the properties of reconstruction error of each macro block (MB) is estimated from the previous frame. For the estimation of the error of the current MB, a block with the size of MB in the previous frame is chosen. Since the original and reconstructed images of the previous frame are available in the encoder, we can evaluate the tendency of reconstruction error of this block in advance. Then, this error is considered as the expected error of the current MB. Comparing the error of the MB with the average of overall MBs, if it is larger than the average, a small step size is given for this MB, and vice versa. As a result, the error distribution of the MB is more concentrated to the average, yielding low variance and improved image quality. For reducing the blocking artifact, the error measure for the blocking artifact is also considered as well as SAD(sum of absolute difference) and PSNR. Especially for low bit application, the proposed algorithm yields much smaller error variance and higher PSNR compared to the conventional TM5. We also propose a modified algorithm for efficient hardware implementation.

I. 서론

MPEG, H.263 등 대부분의 비디오 코딩 표준에

서 영상 블록들은 DCT에 의해 변환되며 계수는 주어진 비트율과 영상의 특성에 따라 적응적으로 양자화된다^[1-4]. 그러나 각 표준안에서는 인코더 각 요

* 서울대학교 전기공학부(nicho@ee.snu.ac.kr)

** 서울시립대학교 제어계측공학과 신호처리연구실(sylee@tina.uos.ac.kr)

논문번호 : 98473-1028, 접수일자 : 1998년 10월 28일

소의 알고리즘을 상세하게 명시하지 않아서 인코더를 설계하는 데 있어서 융통성을 가질 수 있도록 하였다. MPEG-2의 경우 워크 그룹에서 몇몇 인코더 시험 모델을 제안하였으며 여기에 적용 양자화 기법들 또한 소개되어 있다^{5,6)}.

TM5(test model 5)는 거의 표준 MPEG-2 인코더로 인정되어 인코더 구현 및 성능 비교에 이를 기반으로 한 알고리즘들이 널리 사용되고 있다⁶⁾. TM5의 적용 양자화 알고리즘은 매우 간단하면서도 만족스러운 성능을 보인다. [7,8]등에서는 TM5와는 다른 몇 가지 적용 양자화 알고리즘이 제안되었으며, 이들은 영상의 특성을 자세하게 분류하여 인코더의 성능 개선을 시도하는 것이다. 기존의 방법들, 즉 TM5와 [7,8]등의 알고리즘들은 적용 양자화를 위한 정보를 인코딩되는 현재 영상에서만 찾고 있다. 그러나 비디오 코딩 시스템에서는 앞 프레임의 원 영상과 복원 영상이 사용 가능하므로 별도의 프레임 메모리를 추가하지 않고도 이전에 코딩된 영상으로부터 현 프레임의 에러 분포에 대해 더 많은 정보를 얻을 수 있다.

따라서, 본 논문에서는 이전에 코딩된 영상으로부터 얻어진 에러 정보를 이용하는 적용 양자화 알고리즘을 제안한다. 제시되는 비디오 인코더의 개념도가 그림 1에 나타나 있다. 그림 1에서 실선 부분이 기존의 비디오 인코더로서 적용 양자화가 있지만 현재 프레임의 특성만을 이용하는 "feed-forward"방법이라 할 수 있다. 그림 1의 점선 부분이 본 논문에서 추가하는 부분으로서 앞에서 압축-복원된 영상들로부터의 정보를 이용하는 "feed-back"방식이라 할 수 있다.

제안된 알고리즘의 핵심은 이전에 코딩된 영상에서 MB 에러 분포를 조사하여 이 정보를 현재 코딩하는 영상의 에러 분포를 줄이는데 사용하는 것이다. 좀 더 자세히 설명하면, 비교적 큰 에러를 갖는 MB는 제안된 알고리즘에 의해 더 잘게 양자화되고 이로 인해 더 작은 에러를 갖는 MB가 된다. 반대로 더 작은 에러를 갖는 MB는 더 성기게 양자화된다.

일반적으로 복원 에러가 큰 MB에서 영상 왜곡의 가시성이 높아지므로 큰 에러를 줄임으로써 전체적인 화질을 좋게할 수 있다. 본 연구에서는 일반적인 SAD(sum of absolute difference) 외에도 블록 효과를 줄이기 위한 에러 척도를 사용하여 제안된 알고리즘에 대해 다양한 시뮬레이션을 수행해 본 결과, MB들의 에러 분산은 줄고 PSNR 또한 TM5에

비해 개선된다. 본 논문에서는 MPEG-2에 대해서만 제시된 알고리즘을 적용했지만 다른 비디오 코딩 표준안들도 비슷한 인코더 구조를 갖고 있으므로 본 알고리즘은 다른 비디오 코딩 표준에도 쉽게 적용 될 수 있을 것이다.

본 논문은 다음과 같이 구성되어 있다. 2절에서는 본 연구에서 사용한 에러 척도들을 간략하게 소개한다. 3절에서는 제안된 알고리즘을 상세하게 설명한다. 4절에서는 TM5와의 비교 결과를 보이고 보다 효율적인 하드웨어 구현을 위해 수정된 알고리즘도 소개한다.

5절에서는 결론을 맺는다.

II. 복원 에러 척도

본 논문에서는 두가지 에러 척도를 이용한다. 하나는 일반적인 PSNR과 연관된 SAD(sum of absolute difference)이다. 다른 하나는 블록 효과의 에러 측정치인 SAMS(sum of absolute mean difference of slope)로서 [13]에서 제안한 블록 효과 척도인 MSDS(mean square difference of slope)를 수정한 것이다.

본 논문에서는 [13]에서의 기술기 차이의 제곱값 대신에 절대값을 사용한 것이며 블록 단위가 아닌 MB에 대한 에러값을 구하였으며 복원 영상 대신에 원 영상과 복원 영상의 차 영상으로부터 에러값을 구한다.

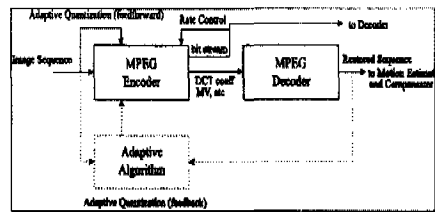


그림 1. 제시된 비디오 인코더의 개념

그림 2는 MB의 위에서 k(=0,1,2,...,15)번째 가로 줄의 각 픽셀의 밝기 값을 그린 것이다. 가로축은 픽셀의 위치를 나타내는 인덱스이며 세로축은 밝기 값을 나타낸다.

MB의 경우 N의 값은 16이다. 그림 2의 두개의 곡선은 블록 효과가 있는 양자화된 영상의 밝기 변화와 블록 효과가 없는 원 영상의 밝기 변화를 보

이고 있다. 양자화된 영상의 경우 블록 경계에서 밝기의 변화가 심하며 이것이 블록 현상으로 보이는 것이다. 따라서 현재의 MB인 (i,j)번째 MB와 이의 왼쪽 MB인 (i,j-1)번째 MB의 경계에서의 밝기의 변화량을 다음과 같은 블록 효과의 척도로 정의한다.

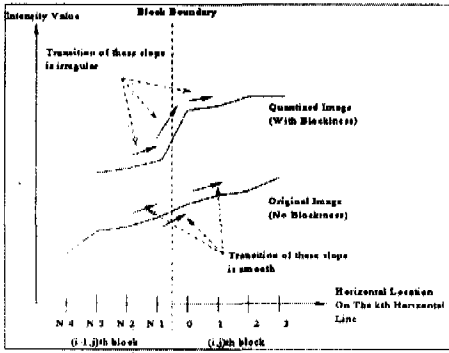


그림 2. MB 경계선에서 픽셀 기울기를 이용한 블록 효과 에러 척도

$$D^L \triangleq \sum_{k=0}^{N-1} |s_{i,j}(k) - \bar{s}_{i,j}(k)|.$$

여기서 $s_{i,j}(k)$ 는 (i,j)번째 MB와 (i,j-1)번째 MB 경계에서의 밝기 값의 기울기를 나타내며

$$s_{i,j}(k) \triangleq x_{i,j}(k,0) - x_{i,j}(k,N-1)$$

로 정의한다. $x_{i,j}(k,n)$ 은 (i,j)번째 MB의 위에서 k 번째 가로줄의 왼쪽에서 n번째 픽셀 값을 나타낸다.

$\bar{s}_{i,j}(n)$ 는 (i,j)번째 MB와 (i,j-1)번째 MB 경계 부근에서의 밝기 값 기울기들의 평균값으로서 다음과 같이 정의하였다.

$$\bar{s}_{i,j}(k) \triangleq [s_{i,j}^H(k,N-1) + s_{i,j}^H(k,1)]/2,$$

$$s_{i,j-1}^H(k,N-1) = x_{i,j-1}(k,N-1) - x_{i,j}(k,N-2),$$

$$s_{i,j}^H(k,1) = x_{i,j}(k,1) - x_{i,j}(k,0).$$

지금까지는 MB의 수직 경계에 대해서 블록 효과 값을 구하였다. 기울기 계산에 이용되는 픽셀들이 위와 아래의 픽셀들이라는 점을 제외하면, 수평 경계에 대해서도 마찬가지로 블록 효과값을 구할 수 있다. 이와 같은 방식으로 그림 3에서 처럼 왼쪽,

오른쪽, 위, 아래 MB와의 경계에 대한 블록효과 값과 점선으로 표시된 현재 MB 자체의 중간 수직·수평 8x8 블록 경계에 대해서도 블록 효과 값을 구하여 이의 합으로 현재 MB에 대한 여러 값을

$$B \triangleq D_{i,j}^L + D_{i,j}^R + D_{i,j}^T + D_{i,j}^B + D_{i,j}^V + D_{i,j}^H,$$

로 정의한다. 여기서 $D_{i,j}^L$ 은 왼쪽 MB와의 경계에서의 블록 효과값을, $D_{i,j}^R$ 은 오른쪽 MB와의 경계에서의 블록 효과값을, $D_{i,j}^T$ 은 위쪽 MB와의 경계에서의 블록 효과값을, $D_{i,j}^B$ 은 아래쪽 MB와의 경계에서의 블록 효과값을, $D_{i,j}^V$ 은 현재 MB 자체의 중간 수직 블록 경계에서의 블록 효과값을, $D_{i,j}^H$ 은 현재 MB 자체의 중간 수평 블록 경계에서의 블록 효과값을 각각 나타낸다.

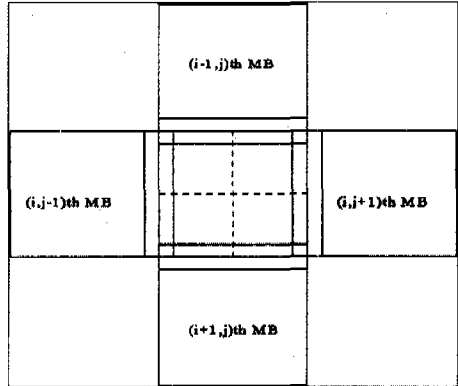


그림 3. 현재의 MB와 블록 효과값 계산에 고려되는 주변 MB들

III. 적응 양자화 알고리즘

TMS^[6]의 적응 양자화 알고리즘은 매우 간단하면서도 만족할 만한 결과를 제공한다. 여기서 양자화 간격의 크기는 블록의 활성도와 비트율에 따라 조정된다. [8]에서는 기존의 양자화 알고리즘의 개선을 위하여 블록을 특성 별로 자세히 분류하여 각각 다른 양자화 간격을 적용하였다. 그러나 기존의 접근 방법들은 인코딩되는 MB의 양자화 에러를 예측하기 위해 현재 프레임 영상의 특성들만을 이용한다.

그러나 모든 비디오 코딩 표준에서는 이전의 원

영상과 복원 영상을 부가 메모리 없이 항상 사용할 수 있으며, 이것은 현 MB의 양자화 에러를 예측하기 위해 이런 영상들에서 유용한 정보를 얻을 수 있다는 것을 의미한다.

즉, 현재 코딩하고자 하는 MB와 특성이 유사한 MB를 이전 프레임에서 얻을 수 있다면 이의 복원 값과 원래값으로부터 현재 MB의 에러 특성을 예측할 수 있다. 이전 프레임에서 비슷한 MB를 찾는 일은 이미 움직임 예측기가 수행을 했으며, 그 결과는 움직임 벡터(MV)로 제공된다. 따라서 움직임 보상된 위치에서 전 프레임의 원 영상과 복원 영상의 차이가 현재 MB의 에러와 유사할 것이라 예측할 수 있다.

그러나 이것이 의미하는 것이 에러의 크기 자체가 이렇게 간단한 방법에 의해 정확하게 예측된다는 것은 아니다. 왜냐하면 비트 할당기가 I, B, 그리고 P 영상에 대해서 각각 다른 비트량을 할당하고 결국 $mquant$ 값⁶⁾이 다르기 때문이다. 그러나 현재 MB가 같은 프레임의 다른 블록들의 에러나 전체 평균 에러보다 더 크거나 더 작은지의 여부, 즉 에러 경향은 예측이 가능하다. 만약 MB의 예측된 에러가 마찬가지로 예측된 전체 에러의 평균보다 크면 세밀하게 양자화 하고, 평균보다 작으면 양자화 간격의 크기를 크게 해서 성긴 양자화를 행한다.

그 결과, 각 MB의 복원 에러의 분포는 에러 평균 주변에 집중된다. 다음 절에서는 다양한 시뮬레이션 결과를 통하여 이와 같은 방법으로 실제로 에러 분산이 줄어들고 많은 경우에 PSNR 값도 개선된다는 것을 보인다.

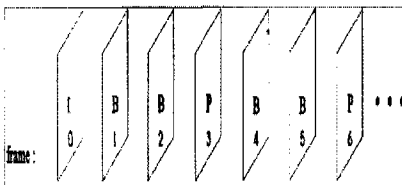


그림 4. MPEG-2에서 영상을 디스플레이하는 순서

MPEG-2의 경우에는 영상의 종류에 I, B, P 영상이 있고 그림 4에서 보는 순서대로 디스플레이된다. 그러나 실제로는 I,P,B,B,P,B,B,P,... 순서로 코딩된다. P 영상을 코딩할 때 이전 영상은 I 혹은 P 영상이다. 따라서 전방향 MV를 이용하여 예측된(움

직임 보상된) 블록을 찾아야 한다.

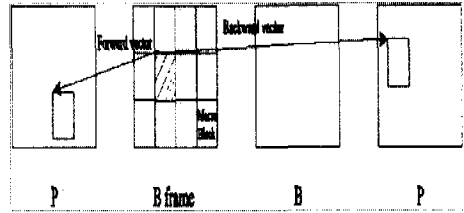


그림 5. B 영상의 순방향·역방향 MV

B 영상의 경우, 그림 5에서 보는 것 처럼 전방향 MV와 후방향 MV를 동시에 사용한다. 움직임 보상을 위해서는 둘 중에 하나만 사용하거나 둘 모두 사용하기도 한다.

그러나 제시된 알고리즘에서 현재 위치와 비슷한 MB를 찾는 작업은 가까운 MV를 사용하도록 하였다. 즉 먼저 코딩되는 B 영상에 대해서는 전방향 벡터를 사용하고 두번째 코딩되는 B 영상에 대해서는 역방향 벡터를 사용하였다. 요약하면, 이전에 코딩된 프레임에서 현재 MB와 유사한 부분을 찾기 위해서는 그림 6에서 보듯이 전방향 혹은 역방향 MV가 사용된다.

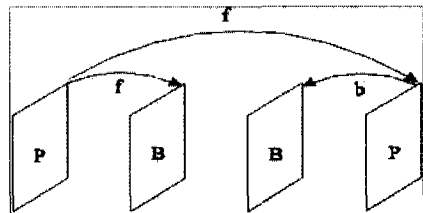


그림 6. MB 복원 에러 계산을 위한 MV

그림에서 “f”는 전방향 벡터를, “b”는 역방향 벡터를 사용한다는 것을 의미한다. I 영상의 경우, 첫 번째 I 영상은 TMS와 같은 방법으로 코딩된다. 나머지 I 영상은 이전의 P 프레임을 이용하여 인코딩된다. 그러나 I 영상에 대해서는 MV가 없으므로 MV는 (0,0)으로 세팅한다. 즉 이전 P 영상에서 같은 위치의 MB를 사용하여 현재 MB의 에러를 예측한다. SAD를 에러 척도로 사용할 경우에는 MV를 이용하여 찾은 전 프레임에서의 MB를 그대로 사용하며 블록 현상을 나타내는 척도인 SAMS를 사용할 경우에는 전 프레임을 코딩할 때의 8x8 블

록 경계들과 일치하지 않고 공간적으로 어긋나 있을 수도 있으므로 일치하도록 조정한다. 이는 어긋나 있는 경우 실제로 블록 경계가 아닌 부분에 대하여 에러값을 구할 수 있으므로 이를 보정하기 위한 것이다.

이를 위해 그림 7에서 처럼 찾은 전 프레임의 MB의 중심점 에서 가장 근접한 매크로 블록이나 블록 경계선을 수직, 수평 방향 세 개씩 찾아 이에 대해 에러값을 구한다. 이 경계선들로 이루어지는 블록은 16x16의 크기를 갖고, 전 프레임에서 찾은 MB와 공간적으로 가장 근접해 있으며 전 프레임에서의 8x8 블록 경계와 일치한다.

인코더의 전체 구조는 TM5와 같고, 그 중에서 양자화 과정은 다음과 같이 정리할 수 있다.

1. 첫번째 I 영상은 TM5와 같은 방법으로 코딩된다.
2. I 혹은 P 영상이 코딩된 후에는 원 영상과 복원 영상으로부터 차 영상을 만든다. 이 영상은 다음 프레임에 대한 MB의 복원 에러를 계산하는데 사용된다. 메모리를 절약하고자 하는 경우에는 차영상을 만들 필요는 없다. 그러나 이 경우에 MB의 복원 에러값은 MB가 참조될 때마다 원 프레임과 복원 프레임으로부터 계산해야 한다.
3. P 영상을 코딩하는 경우에는 순방향 MV가 현재 코딩중인 MB와 가장 유사한 참조 영상에서의 MB를 가리킨다. 이 부분에서 MB 크기의 블록을 차 영상에서 선택하여 에러값을 계산한다. B 영상의 경우, 전방향 혹은 후방향 MV가 사용될 수 있다. 그러나 그림 6에서 보았듯이 가장 근접한 벡터를 사용한다. I 영상의 경우 첫번째 I 영상이 아니면 MV가 (0,0)으로 주어지며 각 MB의 에러값은 이전 P 프레임을 참조하여 계산된다.
4. 알고리즘의 설명을 위하여 각 MB의 에러를 갖고 있는 배열을 $earray[k]$ 라고 가정하자. 물론 배열의 크기는 프레임에서의 MB의 갯수와 같다. $earray[k]$ 는 k번째 MB의 에러 값을 갖고 있다. [6]에서와 같이 양자화 스텝 크기를 $mquant$ 라고 할 때 본 논문의 적응 양자화 알고리즘의 주요 부분은 다음과 같다.

```
for(j = 0; j < B; j++) {
```

```
    Compute mquant as in TM5;
```

```
    ratio = earray[k]/average;
```

```
    mquant = mquant/ratio;
```

```
    Clipping and refinement of mquant as in TM5;
```

```
}
```

여기서 B는 프레임의 MB 갯수이고, average는 MB 에러들의 평균으로서 다음과 같다.

$$average = \frac{1}{B} \sum_{k=0}^{B-1} earray[k].$$

위의 알고리즘에서 MB의 현재 에러값이 평균보다 크면 ratio는 1보다 커진다. 그러므로, TM5의 알고리즘으로 계산된 mquant는 ratio로 나누어짐으로써 값이 작아진다.

MPEG-2 표준에서는 mquant는 1에서 31사이의 값을 가지며 작은 mquant값은 세밀한 양자화를 의미한다^[6]. 반대로 평균보다 작은 에러값을 갖는 MB에 대해서는 ratio는 1보다 작고 따라서 전보다 더 성기게 양자화 된다.

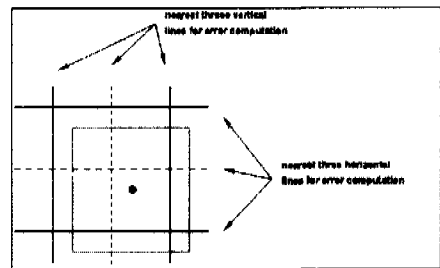


그림 7. SAMS가 에러 최소인 경우 에러 계산을 위한 블록 경계를 찾는 방법

IV. TM5와의 비교

제안된 알고리즘의 성능을 TM5와 비교하기 위해 각각 다른 특성을 갖는 몇가지 이미지 시퀀스를 다른 비트율로 코딩해 보았다. 본 논문에서의 “분산”은 MB의 복원 에러값의 분산이며 다음처럼 정의한다.

$$variance = \frac{1}{B} \sum_{k=0}^{B-1} (earray[k] - average)^2.$$

물론, 더 작은 분산값은 MB의 복원 에러가 다른

MB들과 많은 차이가 나지 않는다는 것을 의미한다. 제안된 방법을 TM5의 성능과 비교하기 위해 popple, akiyo, football 시퀀스에 대해서 다음과 같은 조건하에 실험하였다.

- 비트율은 2.5, 6Mbps 두 가지 경우를 고려한다.
- 사용된 영상은 720x480의 크기를 갖는다.
- M=3, N=15인 GOP 구조를 갖도록 한다.
- Frame DCT와 Frame ME를 행하도록 한다.

그림 8에서 그림 10까지 제안된 알고리즘과 TM5의 SAD 에러 분산과 PSNR을 각각의 경우에 대해서 나타내었다. 그림 11에서 그림 13까지는 SAMS의 경우를 나타내었다. 표 1에는 에러 척도가 SAD 일 때의 120 프레임에 대한 생성된 비트 스트림의 크기, 에러 분산의 평균, PSNR의 평균 값을 나타내었다. 결과를 보면 제안된 알고리즘이 일반적으로 비트 스트림의 크기는 TM5의 것과 거의 비슷하면 서도 더 나은 화질의 영상을 복원한다는 것을 알 수 있다. 이것은 또한 제안된 알고리즘이 TM5의 비트율 제어 알고리즘에서 잘 동작한다는 것을 의미한다. SAMS를 척도로 사용한 경우도 비트스트림 크기는 거의 같았고 매우 빠른 움직임이 있는 football 영상 외에는 블록현상이 줄었고 PSNR도 개선되었다.

그림 8에서 그림 10까지, 표 1, 그리고 그림 11에서 그림 13까지로부터 제안된 알고리즘은 특히 낮은 비트율과 느린 움직임의 영상에서 TM5에 비해 좋은 결과를 보이는 것을 알 수 있다. 결과에서 주목할 점은 PSNR이 높아졌다는 것이 아니라 에러 분산이 줄었다는 것이다. 왜냐하면 제안된 알고리즘은 영상 전체 에러를 줄이기 보다는 작은 에러를 갖는 MB이미지의 화질 열화를 감수하고 평균보다 큰 에러를 갖는 MB의 에러를 줄이고자 했기 때문이다. 에러 분산을 줄임으로써 큰 에러를 갖는 MB들은 작은 에러 값을 갖게 된다. 이것은 그림 14에서 보다 시각적으로 알아 볼 수 있는데, 이는 TM5와 제안된 알고리즘에 의해서 만들어진 복원 영상 및 에러 영상이다. 에러 영상은 차영상이 아니라 SAD를 MB 단위로 밝기로 나타낸 것이다. 그림 14에서 더 밝은 블록은 더 큰 에러를 갖는 블록이다. 그림 15은 TM5에 의한 영상과 제시된 알고리즘에서 SAMS를 사용한 결과 영상을 비교한 것이다. 이로부터 SAMS를 사용하면 블록 현상이 줄어들을

알 수 있다.

일반적으로 높은 복원 에러값을 갖는 MB에서 영상 왜곡이 보다 심하므로 큰 에러 분산을 줄임으로써 전면적으로 영상의 왜곡을 줄일 수 있다. 그러나 밝기, 주파수, 에지, 움직임^[9-12]과 같은 왜곡의 가시도에 영향을 미치는 많은 부수적인 요인들이 있기 때문에 SAD나 단순한 블록 현상의 측정치인 SAMS값만을 줄이려 하기 보다는 주변의 주파수와 밝기등의 마스킹 효과를 고려하는 인간의 시각 감각에 바탕을 둔, 여러 측정 방법도 고려해 보아야 하며 추후 연구 주제로 남아 있다.

제안된 알고리즘은 매우 간단하므로 소프트웨어, 하드웨어 구현에 별 어려움이 없다. 그러나 전체적인 하드웨어가 MB 단위로 동작을 하는 경우에는 제안된 알고리즘을 사용하기 곤란하다. 즉 MV 탐색에서 비트 스트림 출력에 이르기 까지 한 MB 씩 수행되는 경우 제안된 알고리즘을 이용할 수 없다. 왜냐하면 MB 에러의 평균은 MB를 인코딩하기 전에 계산되어야하고 MB 에러의 계산에는 MV가 필요하기 때문이다. 한 MB 씩 이 알고리즘을 적용하려면 약간의 성능 저하를 감수하고 알고리즘을 더욱 간략화 할 필요가 있다. 이를 위하여 수정한 알고리즘에서는 움직임 예측기로부터의 정확한 MV를 사용하지 않고 MV를 단순히 (0,0)으로 한다. 즉 각 MB의 에러값은 이전 프레임의 같은 위치에서 구한 값이다. 이렇게 하면 에러의 평균을 MB에 대한 양자화를 시작하기 전에 구할 수 있고 모든 과정이 한MB 씩 수행되게 된다. Popple 영상에 대해 제안된 알고리즘, 이의 수정된 알고리즘 및 TM5를 비교한 것을 그림 16에 보였다. 그림 16에서 수정된 알고리즘이 원래의 알고리즘과 비교할 때 약간의 성능 저하가 있지만 TM5보다는 역시 더 좋은 결과를 보이고 있다는 것을 알 수 있다. 그림 17은 좀 더 빠른 움직임을 갖는 영상(football)에 대해서 비교해 본 것이다. 통계적인 결과 또한 표 2에 TM5와 원래의 알고리즘과 비교하여 요약되어 있다. 이 결과 또한 수정된 알고리즘의 PSNR은 거의 같으나 여전히 훨씬 낮은 에러 분산을 낸다는 것을 보이고 있다. 각 알고리즘에 대해서 실제 발생한 비트 스트림의 크기는 거의 같았다.

일반적으로 TM5 뿐만 아니라 제안된 인코더도 더 큰 MV 탐색 영역에서 더 좋은 결과를 보인다. 제안된 인코더의 경우 ± 63 의 탐색 영역을 가질 때 ± 31 혹은 ± 15 일 경우 보다 일반적으로 결과가 더 좋았으며 TM5에 비해서는 탐색 영역이 클 때 개선

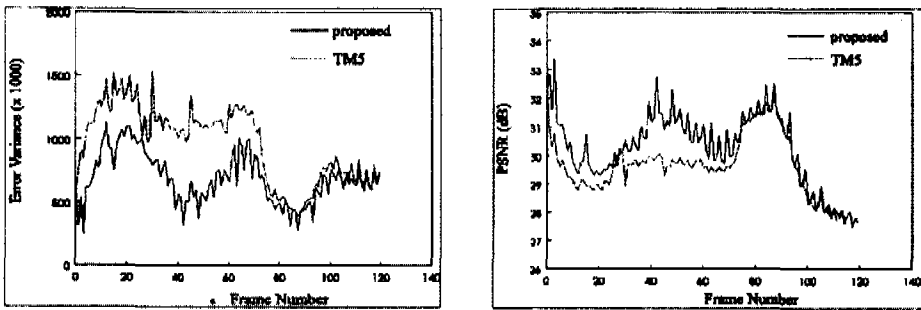
효과가 더 현저했다.

V. 결론

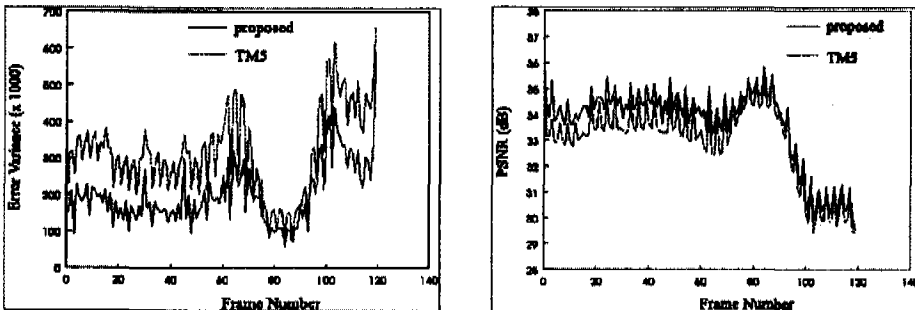
본 논문에서는 비디오 코딩을 위한 적응 양자화 알고리즘을 제안하였다. 기존의 적응 양자화 방법은 현재 영상의 특성만을 이용하는 반면에, 제시된 방법에서는 양자화기의 스텝 크기는 이전에 코딩된 영상으로부터 얻어진 정보를 이용하여 조정된다. 제안된 알고리즘에서 사용된 모든 정보는 별도의 프레임 메모리 없이 이미 비디오 코딩 표준에서 얻을 수 있는 값들이기 때문에 알고리즘의 구현은 매우 간단하다. 다양한 시뮬레이션을 통해 제안된 알고리즘에 의한 에러 분산은 거의 모든 경우에 있어 TMS의 경우보다 더 작음을 알 수 있었다. PSNR도 움직임이 적은 영상과 낮은 비트율 코딩에서 많이 개선되었다. 이것은 매우 큰 복원 에러를 갖는 MB들은 제안된 알고리즘에 의해 더 작은 에러를 갖게 된다는 것을 의미한다. 또한 보다 효율적인 하드웨어 구현을 위해 수정된 알고리즘도 제안하였다.

TMS는 갑자기 변하는 장면과 VBV(Video Buffering Verifier)의 요구사항을 따르는데 한계가 있다는 것이 알려져 있다^[6,14]. 제안된 비디오 인코더는 TMS를 바탕으로 하므로 마찬가지로 문제점을 안고 있다. 예를 들어 제안된 알고리즘이 장면 전환 후에 수행될 때 이전 프레임에서의 에러 예측치는 잘못된 값이다. 이러한 문제는 제안된 양자화 기술을 장면 전환 탐지 알고리즘과 함께 사용함으로써 정감되리라고 예상된다. 예를 들어 장면 전환이 [14,15]의 알고리즘에 의해 탐지될 때 스텝 크기를 바꾸는 과정은 생략할 수 있다.

본 논문에서는 에러 측정 방법으로 SAD와 블록 현상 감소를 위한 SAMS 두개 만을 사용하였다. 만약 인간의 시각 감각 및 움직임의 마스킹 효과^[9-12]를 고려한 에러 측정기를 사용한다면 주관적인 영상의 화질은 더 개선될 것이라고 예상된다. 또한 본 논문에서의 알고리즘을 MPEG-2에만 적용하였으나 다른 비디오 코딩 표준에서도 이를 쉽게 구현할 수 있을 것이다.

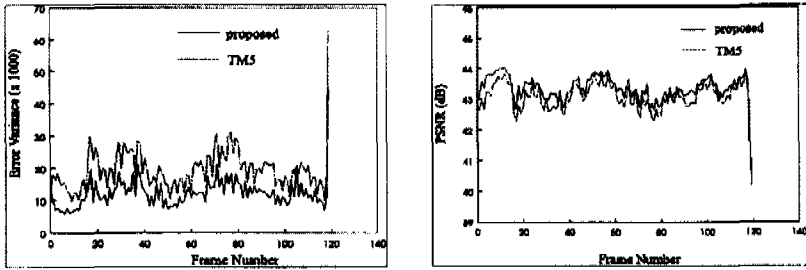


(a) 2.5 Mbps

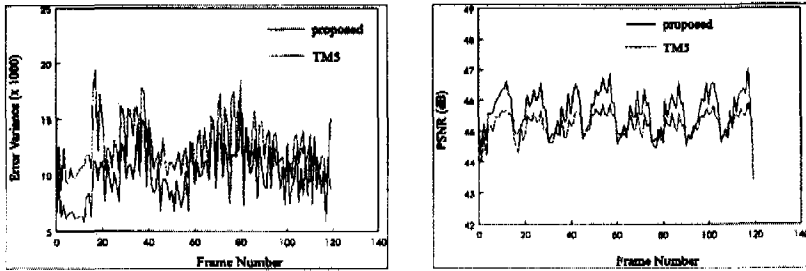


(b) 6.0 Mbps

그림 8. popple 영상의 SAD 에러 분산 및 PSNR

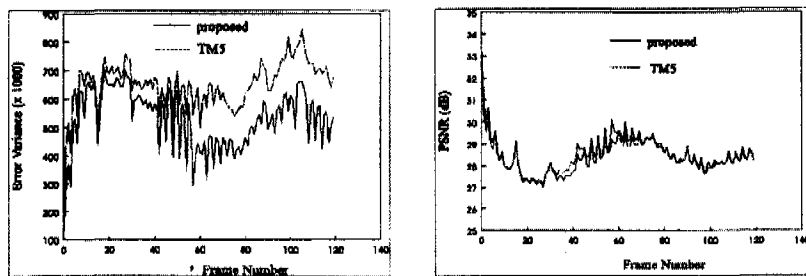


(a) 2.5 Mbps

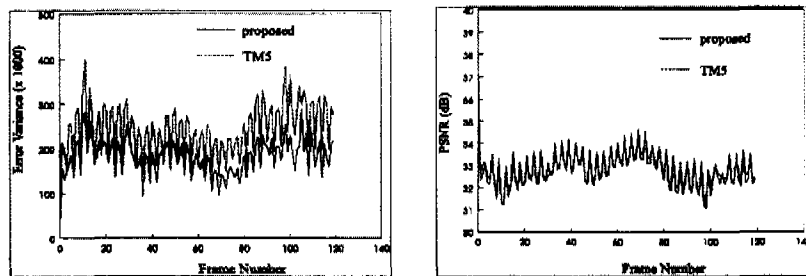


(b) 6.0 Mbps

그림 9. akiyo 영상의 SAD 에러 분산 및 PSNR

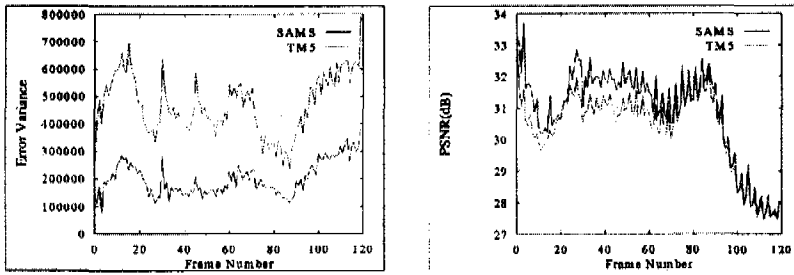


(a) 2.5Mbps

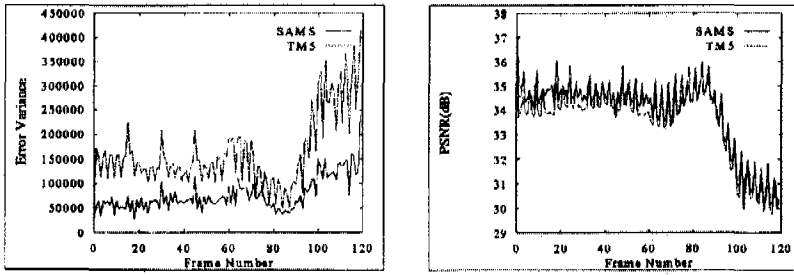


(b) 6.0Mbps

그림 10. football 영상의 SAD 에러 분산 및 PSNR

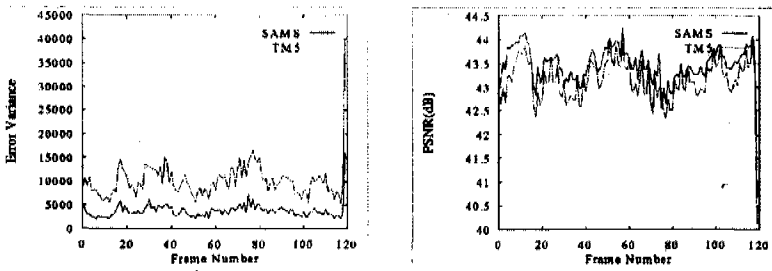


(a) 2.5Mbps

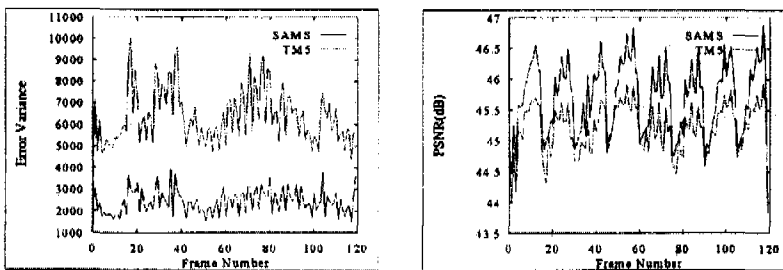


(b) 6.0Mbps

그림 11. popple 영상의 SAMS 에러 분산 및 PSNR

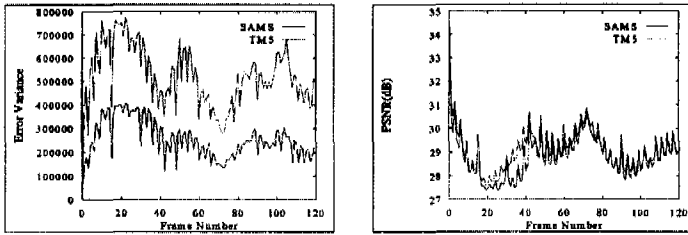


(a) 2.5 Mbps

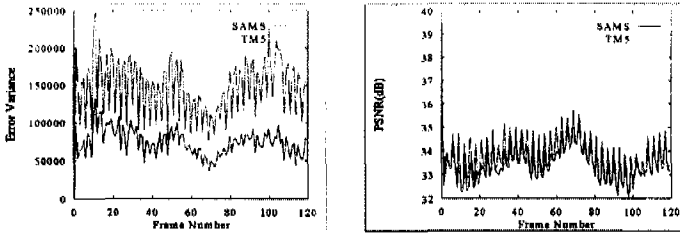


(b) 6.0Mbps

그림 12. akiyo 영상의 SAMS 에러 분산 및 PSNR

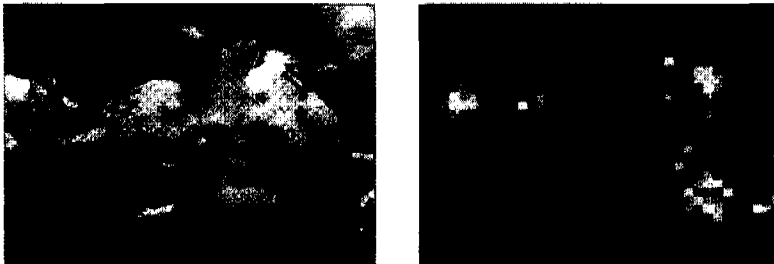


(a) 2.5Mbps

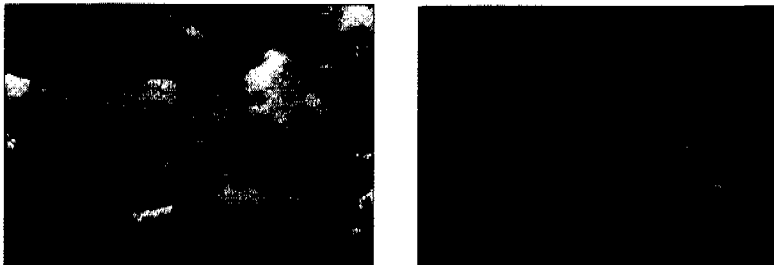


(b) 6.0Mbps

그림 13. fooball 영상의 SAMS 에러 분산 및 PSNR



(a) TM5의 경우 복원된 영상과 SAD에러 영상



(b) 제한된 알고리즘의 경우 복원된 영상과 SAD 에러 영상

그림 14. TM5와 제한된 알고리즘의 복원된 영상과 에러 영상-20번째 프레임

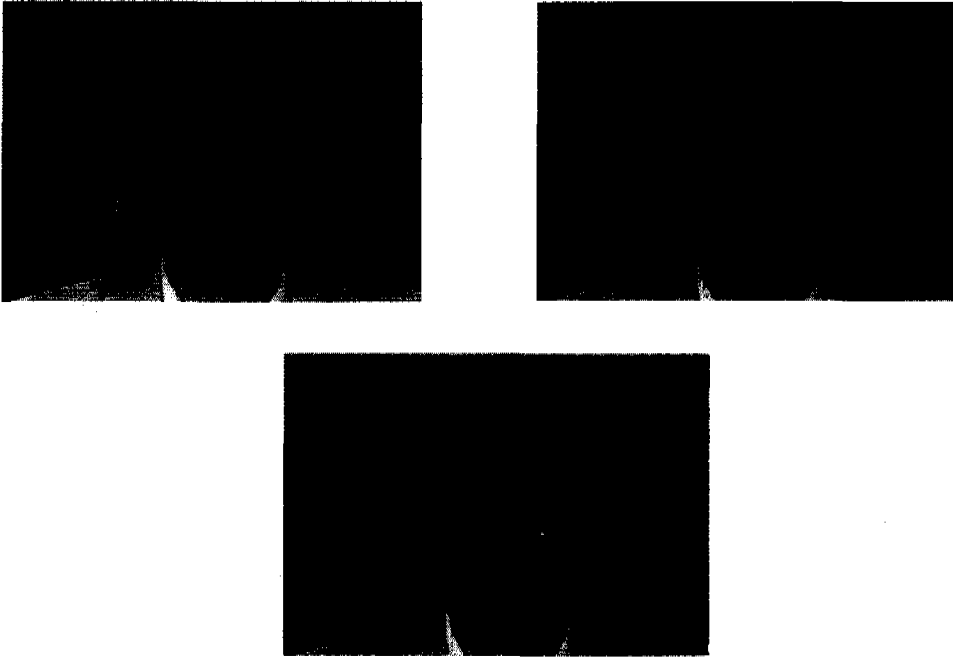


그림 15. TM5와 제안된 방법에 의한 블록 효과의 비교(위에서부터 원영상, TM5에 의한 영상, SAMS에 의한 영상)

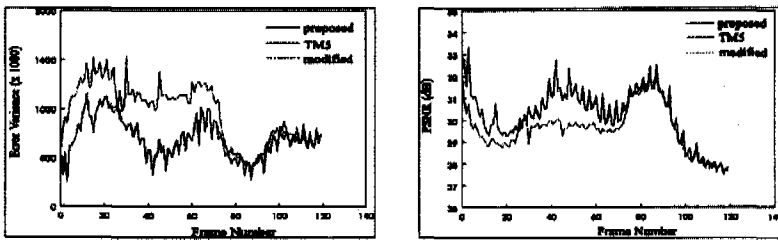


그림 16. popple 영상(2.5Mbps)의 경우 제안된 알고리즘과 TM5와 수정된 알고리즘에 대한 SAD 에러 분산과 PSNR의 비교

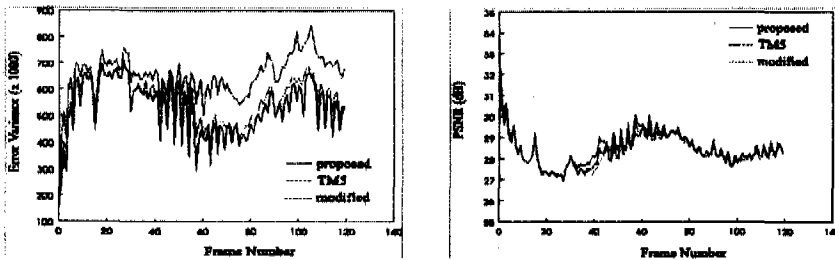


그림 17. football 영상(2.5Mbps)의 경우 제안된 알고리즘과 TM5와 수정된 알고리즘에 대한 SAD 에러 분산과 PSNR의 비교

표 1. 102 프레임에 대한 시물레이션 통계치(에러 척도: SAD)

영상	비트율	실제비이트수		에러 분산의 평균		PSNR의 평균	
		TM5	제안된 방법	TM5	제안된 방법	TM5	제안된 방법
Popple	2.5	1,299,295	1,299,917	965,439	690,999	29.62	30.27
	6.0	3,005,113	3,005,323	314,928	201,020	33.02	33.64
Akiyo	2.5	1,254,461	1,254,688	19,506	12,249	43.08	43.34
	6.0	3,001,520	3,001,772	12,469	9,949	45.18	45.61
Football	2.5	1,257,526	1,258,284	650,471	527,318	28.45	28.48
	6.0	3,007,283	3,007,130	242,575	181,424	32.82	32.89

표 2. 수정된 알고리즘을 이용하여 2.5Mbps의 비트율로 코딩된 popple과 football 영상에 대한 통계치(에러 척도: SAD)

영상	에러분산의 평균			PSNR의 평균		
	TM5	수정된 방법	제안된 방법	TM5	수정된 방법	제안된 방법
Popple	965,439	707,773	690,999	29.62	30.21	30.27
Football	650,471	555,559	527,318	28.45	28.36	28.48

참 고 문 헌

[1] Motion Picture Expert Group, MPEG Draft International Standard, Technical Report, CCITT SGXV, March 1994.

[2] D. Le Gall, "MPEG : A video compression standard for multimedia applications," *Commun. ACM*, vol. 34, pp. 47 - 58, April 1991.

[3] ISO/IEC JTC1/SC29/WG11, "Coding of moving pictures and associated audio information - MPEG-4 video verification model version 3.1," August 1996.

[4] International Telecommunication Union, Line Transmission of Non-telephone Signals, Draft ITU-T Recommendation H.263, April 1995.

[5] Motion Picture Experts Group, "Description of simulation model 3," MPEG 90/041, July 1990

[6] ISO/JTC1/SC29/WG11/N0400, "Coded representation of picture and audio information - MPEG2 Test Model 5," April 1993.

[7] E. Viscito, C. Gonzales, "A video compression algorithm with adaptive bit allocation and quantization," *SPIE Vol. 1605 Visual Communication and Image Processing*, '91 Visual Communication, pp. 58 - 72, 1991.

[8] A. Puri and R. Aravind, "Motion-compensated video coding with adaptive perceptual quantization," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 1, no. 4, pp. 351 - 361, December 1991.

[9] S. A. Karunasekera, N. G. Kingsbury, "A distortion measure for blocking artifacts in images based on human visual sensitivity," *IEEE Trans. on Image Processing*, vol. 4, no. 6, pp. 713 - 724, June 1995.

[10] K. N. Ngan, "Hybrid image coding scheme incorporating human visual system characteristics," *Optical Engineering*, vol. 30, pp. 940 - 946, July 1991.

[11] N. Li, S. Desmet, A. Deknuydt, L. V. Eycken, "Motion adaptive quantization in transform coding for exploiting motion masking effect," *SPIE Vol. 1818 Visual Communication and Image Processing '92*, pp. 1116 - 1123, 1992.

[12] S. K. Paek, J. S. Kang, Y. S. Seo, "Rate control strategy based on human visual sensitivity for MPEG video coder," *SPIE Vol. 2308 Visual Communication and Image Processing '94*, pp. 322 - 330, 1994.

[13] Shigenobu Minami and Avidesh Zakhor, "An optimization approach for removing blocking effects in transform coding," *IEEE Trans. Circuits*

