

論文2003-40SP-5-13

H.264의 가변 블록 크기 움직임 추정 및 공간 예측 부호화 생략에 의한 고속 모드 결정법

(Fast mode decision by skipping variable block-based motion estimation and spatial predictive coding in H.264)

韓基勳*, 李英烈*

(Ki-Hun Han and Yung-Lyul Lee)

요약

ITU-T(International Telecommunication Union-Telecommunication standardization sector)와 MPEG(Moving Picture Experts Group)에 의해서 최근 표준화가 완성된 H.264는 가변 블록 크기 움직임 추정, 복수참조영상, 1/4화소 움직임 예측/보상, 4×4 정수 DCT(Integer Discrete Cosine Transform), 율-왜곡 최적화(Rate-Distortion Optimization) 등의 새로운 부호화 기술로 H.263, MPEG-4 등 기존 비디오 표준에 비해 더 좋은 부호화 효율을 제공하고 있다. 그러나 새로운 부호화 기술들은 H.264 의 전반적인 복잡도를 심화시키는 주된 요인이기도 하다. 따라서, H.264 의 실제 응용을 용이하게 하기 위해서는 이러한 기술에 대한 고속 알고리즘이 요구된다. 본 논문에서는 율-왜곡 최적화를 통한 부호화 모드 결정시 부호화기의 복잡도에서 가장 큰 비중을 차지하는 가변 블록 크기 움직임 추정 및 공간예측 부호화를 효율적으로 생략하여 부호화 모드 결정을 빠르게 수행하는 고속 모드 결정법을 제안한다. 실험결과, 제안된 방법은 부호화 효율의 손실이 거의 없으면서도 계산량을 약 4배 향상 시킨다.

Abstract

H.264, which is the latest video coding standard of both ITU-T(International Telecommunication Union-Telecommunication standardization sector) and MPEG(Moving Picture Experts Group), adopts new video coding tools such as variable block size motion estimation, multiple reference frames, quarter-pel motion estimation/compensation(ME/MC), 4×4 Integer DCT(Discrete Cosine Transform), and Rate-Distortion Optimization, etc. These new video coding tools provide good coding efficiency compared with existing video coding standards as H.263, MPEG-4, etc. However, these new coding tools require the increase of encoder complexity. Therefore, in order to apply H.264 to many real applications, fast algorithms are required for H.264 coding tools. In this paper, when encoder MacroBlock(MB) mode is decided by rate-distortion optimization tool, fast mode decision algorithm by skipping variable block size ME/MC and spatial-predictive coding, which occupies most encoder complexity, is proposed. In terms of computational complexity, the proposed method runs about 4 times as fast as JM(Joint Model) 42 encoder of H.264, while the PSNR(peak signal-to-noise ratio)s of the decoded images are maintained

Keyword : Motion Estimation, variable block, H.264, Rate distortion, Mode decision

* 學生會員, 世宗大學校 인터넷工學科
(Department of Internet Engineering Sejong Uni-

versity)

接受日字:2003年8月27日, 수정완료일:2003年9月15日

I. 서론

H.263^[1], MPEG-4^[2]와 H.264^[3] 등의 영상 압축 표준은 블록기반의 움직임 추정(motion estimation)을 수행한다. 움직임 추정은 동영상 부호화에서 시간적 중복성(temporal redundancy)을 제거하여 비트율(bit-rates)을 줄이는데 큰 역할을 한다. 특히, H.264는 다양한 크기의 가변 블록 크기 움직임 추정(variable block-based motion estimation)을 하여 부호화 효율에 있어 많은 이득을 제공한다. 또한 H.264는 1/4화소 단위의 움직임 벡터 예측을 수행하여 더욱 정확한 움직임 추정으로 부호화 효율을 한층 향상 시켰다. 그리고 복수개의 참조영상(multiple reference frames)을 지원하기 때문에 현재 블록과 중복성이 가장 많은 프레임의 블록을 참조영상으로 선택하여 이전 프레임만을 참조영상으로 사용할 때 보다 더욱 큰 부호화 효율을 얻을 수 있었다. 또한 움직임 추정에 쓰이는 가변블록 모드와 3가지의 공간예측 모드(Intra16×16, Intra4×4, IBLOCK), SKIP모드 등 모든 가능한 부호화 모드 중에서 가장 최적의 모드를 선택하기 위해 윌-왜곡 최적화^[4-5] 기술을 사용하여 H.264 BP(Baseline Profile)의 부호화 효율을 더욱 향상 시켰다. 이러한 새로운 부호화 기술로 인해, H.264는 기존의 H.263이나 MPEG-4 SP(Simple Profile)에 비해 동일한 화질에서 압축률이 최대 1.5 ~ 2 배 가량 더 좋다는 것이 실험을 통해 밝혀졌다^[6-7]. 그러나 새로운 부호화 기술을 채택함으로써 연산량이 증가하여 복잡도면에서는 MPEG-4 SP 부호화기 보다 약 16배 높은 것으로 나타났다^[8]. 따라서 H.264의 실제 응용을 용이하게 하기 위해서는 이러한 복잡도를 줄여주는 고속 알고리즘의 개발이 필요하다. 본 논문에서는 H.264의 부호화 시간 중 가장 큰 비중을 차지하는 가변 블록 크기 움직임 추정 및 공간예측 부호화를 효율적으로 생략하여 윌-왜곡 최적화를 통한 부호화 모드 결정에서의 속도향상을 위한 고속 모드 결정법을 제안한다. 본 논문의 II장에서는 H.264 BP의 부호화 과정을 서술하고, III장에서는 부호화 속도향상을 위한 고속 모드 결정법을 제안한다. 제안된 방법의 성능은 IV장에서 실험결과를 통해 확인하겠으며 마지막으로 V장에서 본 논문의 결론을 내리고자 한다.

II. H.264 표준의 윌-왜곡 최적화에 의한 부호화 모드 결정

H.264의 부호화 효율을 향상하기 위한 가변 블록 크기 움직임 추정에 사용되는 7가지 가변 블록은 <그림 1>과 같다. H.264는 움직임 추정시 하나의 16×16 매크로 블록을 16×16, 16×8, 8×16, 8×8블록으로 나누고 다시 8×8블록을 8×4, 4×8, 4×4블록으로 나누어 가변 블록으로 움직임 추정을 수행한다. 이러한 다양한 가변 블록 크기의 움직임 추정은 영상의 특성과 움직임에 따라서 효과적으로 부호화 할 수 있다. 영상의 움직임이 적고 물체가 클 경우 큰 블록크기로, 영상의 움직임이 복잡하고 물체가 작을 경우 작은 블록크기로 움직임 추정 및 부호화 하는 것이 효과적이다. 가변 블록 크기의 움직임 수행 후, 가변블록 모드와 3가지의 공간 예측 모드, SKIP 모드 등 모든 가능한 부호화 모드 중에서 최적의 모드를 선택하기 위해 윌-왜곡 최적화를 수행하여 RDcost가 최소가 되는 모드를 부호화 모드로 결정한다. 각 모드별로 화질의 열화(Distortion)와 비트량(Rates)을 고려한 RDcost의 계산식은 식 (1)과 같다.

$$RD\ cost = Distortion + \lambda_{Mode} \times Rates \quad (1)$$

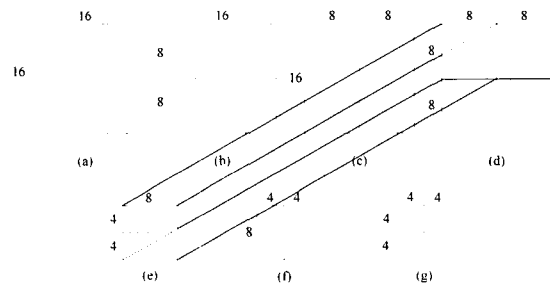


그림 1. H.264 표준의 움직임 추정에 사용되는 7가지 가변블록
Fig. 1. Variable blocks used in motion estimation in H.264.

식 (1)에서 Distortion은 현재 영상과 복원된 영상의 error값이며 식 (2)와 같이 계산된다.

$$Distortion = \sum_{k=0}^{15} \sum_{l=0}^{15} (B(k,l) - B'(k,l))^2 \quad (2)$$

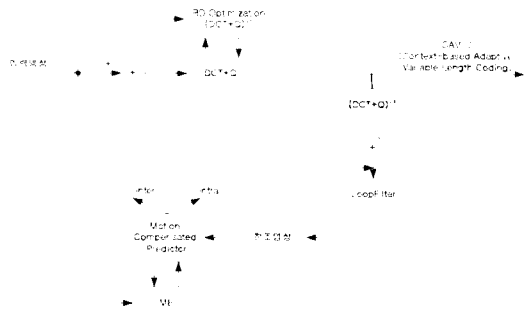


그림 2. H.264 부호화기
Fig. 2. H.264 encoder.

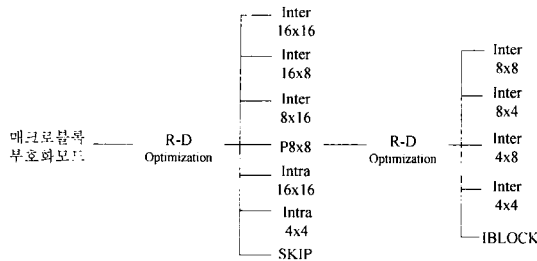


그림 3. H.264의 부호화 모드 결정
Fig. 3. Encoding mode decision in H.264.

식 (2)에서 $B(k,l)$ 과 $B'(k,l)$ 은 각각 현재 매크로 블록과 복원된 매크로 블록의 (k,l) 번째 화소값을 나타낸다. λ_{Mode} 는 Lagrangian 상수로서 식 (3)과 같이 구할 수 있다^[9].

$$\lambda_{Mode} = 0.85 \times 2^{(QP-12)/3} \quad (3)$$

식 (3)에서 QP는 0에서 51까지의 값을 가질 수 있는 H.264 양자화 값(Quantization value)이다. <그림 2>와 <그림 3은> H.264의 부호화기와 율-왜곡 최적화에 의한 부호화 모드 결정을 나타낸다. <그림 3>에서 볼 수 있듯이 매크로 블록의 부호화 모드는 Inter16×16, Inter16×8, Inter8×16, P8×8, Intra16×16, Intra4×4, SKIP 모드 등 7가지가 있다. SKIP모드는 (0,0)벡터 또는 Inter16×16 모드의 움직임 벡터를 가지며 DCT 및 양자화 후 잔여신호(residual signal)가 모두 0인 경우이다. P8×8모드는 매크로 블록내의 4개의 8×8블록들이 독립적으로 율-왜곡 최적화를 거쳐 Inter8×8, Inter8×4, Inter4×8, Inter4×4, IBLOCK 등 5가지 모드 중 최적모드로 부호화 된다. IBLOCK 모드는 8×8블록을 Intra4×4모드로 부호화 하는 것이다. Intra4×4모드는

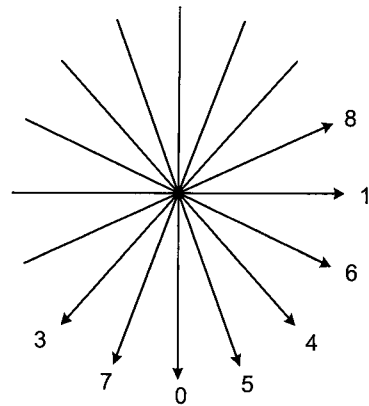


그림 4. Intra44 예측 부호화시 9가지 예측모드
Fig. 4. 9 types of intra4x4 predict mode.

<그림 4>에서 볼 수 있듯이 9가지 예측방향에 대해서 모두 RD_{cost} 를 계산한다. 식 (1)로부터 RD_{cost} 를 계산하기 위해서는 4×4 정수 DCT+양자화, Entropy 부호화, 복호화, (DCT+양자화)¹가 수행되어야 함을 알 수 있다. DCT+양자화, Entropy 부호화, 복호화, (DCT+양자화)¹가 4×4블록단위로 이루어지므로 매크로 블록 단위에 P8×8모드와 Inter16×16모드의 RD_{cost} 계산 수행횟수를 비교하면 Inter16×16모드는 16(16개의 4×4블록)번인 반면 P8×8 모드는 4(4개의 8×8블록) 4(4개의 4×4블록) 4(Inter8×8, Inter8×4, Inter4×8, Inter4×4) + 4(4개의 8×8블록) 4(4개의 4×4블록) 9(9개의 예측모드) = 208번으로 무려 13배의 복잡도를 가진다. 모든 부호화 가능한 모드들에 대한 RD_{cost} 를 계산하는 것은 많은 계산량을 요구한다. 따라서 매크로 블록의 부호화 모드를 조기에 효과적으로 결정할 수 있다면 불필요한 RD_{cost} 계산 및 움직임 추정을 생략하여 전체적인 부호화기의 복잡도와 계산량을 감소시킬 수 있을 것이다. 가변 블록 크기의 움직임 추정과 공간예측 부호화가 부호화 효율 및 부호화 시간에 미치는 영향을 확인하기 위한 실험을 수행하였다. <표 1>에서 H.264는 7가지 가변 블록 크기의 움직임 추정 및 공간예측 부호화를 수행하여 부호화 한 결과이고, Simple H.264는 H.264와 같은 조건에서 7가지 가변 블록 크기의 움직임 추정과 공간예측 부호화를 사용하지 않고 16×16 블록단위의 움직임 추정만을 사용하여 부호화 한 결과이다. H.264와 Simple H.264에 공통으로 사용된 부호화 조건은 다음과 같다. 30frames/sec으로 저장된 300장의 영상에 대하여 10frames/sec의 10Hz프레임율로 영상을

압축하면서, 첫 번째 프레임만 Intra 프레임으로 나머지는 모두 Predictive 프레임으로 부호화 하였다. 1장의 참조영상을 사용하였으며, 16의 검색영역, 1/4 화소 단위의 움직임 예측/보상 그리고 하다마드 변환, (4×4정수 DCT + Q) 계수에 대한 CAVLC(Context-based Adaptive Variable Length Coding)^[10-11]를 이용하여 부호화 하였다. 실험에 사용된 영상은 Foreman QCIF(176×144) 영상이며 QP(Quantization Parameter) 값을 각각 28, 32, 36, 40으로 부호화 하였다. <그림 5>에서 H.264와 Simple H.264의 부호화 시간과 PSNR을 확인할 수 있다. <표 1>은 QP 28일때 Simple H.264와 H.264의 성능을 수치로 나타냈다. Simple H.264는 H.264에 비해 PSNR은 0.49dB 떨어지고 비트율은 22.9% 증가한다. 그러나 전체적인 부호화 시간은 H.264에 비해 48.3%만이 소요되는걸 알 수 있다. 이 결과로 볼 때 가변 블록 크기의 움직임 추정과 공간예측 부호화는 부호화기의 성능에 미치는 영향이 크고 또한 그에 따른 복잡도 역시 매우 높음을 알 수 있다. 이에 본 논문에서는 부호화기의 성능을 유지하면서 속도를 향

표 1. Simple H.264와 H.264의 압축효율 및 부호화 시간 비교

Table 1. Comparison of coding efficiency and encoding time between Simple H.264 and H.264.

QP=28	PSNR(dB)	Bitrates(Kbps)	Encoding time(%)
Simple H.264	35.29	94.96	48.3
H.264	35.78	77.27	100.0

상시키기 위해 부호화기에서 가장 많은 계산량을 차지하는 가변 블록 크기 움직임 추정 및 공간예측 부호화를 효율적으로 생략하여 울-왜곡 최적화를 통한 부호화 모드 결정시 속도를 향상 시켜주는 고속 모드 결정법을 제안한다.

III. 제안된 고속 모드 결정법

본 장에서는 가변 블록 크기 움직임 추정과 공간예측 부호화를 효율적으로 생략하는 고속 모드 결정법을 제안한다.

1. 가변 블록 크기의 움직임 추정에서의 고속 모드 결정법

본 단락에서는 움직임 추정과 RD_{cost} 를 구하는 과정에서 많은 계산량을 요구하는 P8×8모드에 대한 움직임 추정과 RD_{cost} 계산을 효과적으로 skip하는 방법을 제시한다. 불필요한 움직임 추정 및 울-왜곡 계산을 skip하기 위하여 가변 블록 크기의 움직임 추정을 <그림 6>과 같은 순서로 수행하였다. 가변 블록 크기의 움직임 추정에서 매크로 블록이 작은 블록들로 쪼개어지면 움직임 추정을 한다면 16×16블록으로 움직임 추정을 하였을 때 보다 더 자세한 움직임을 표현 할 수 있어 $Distortion$ 은 작아지지만 각 블록에 대한 움직임벡터, 블록코딩패턴(Coded Block Pattern), 부호화모드 등 추가적인 정보들로 인해 비트율이 증가 할 수 있다. RD 측면에서 매크로 블록이 여러방향으로 움직임을 갖고 있어 블록이 나누어질 때 $Distortion$ 이 크게 감소한다면 계속해서 더 작은 블록들로 나누어져서 움직임 추정을 수행하여야 한다. 그러나 블록이 나누어졌을 때 감소하는 $Distortion$ 보다 증가하는 비트율이 더 많다면 큰 블록모드를 유지하는 것이 바람직하다. 제안되는 방법은 Inter16×16, Inter16×8, Inter8×16모드의 움직임 추

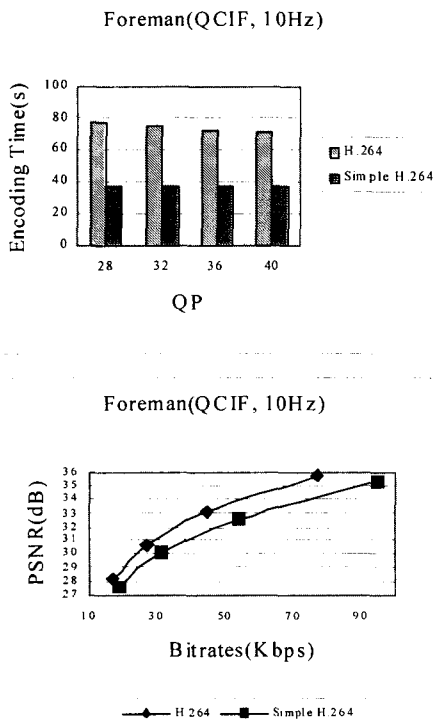


그림 5. H.264와 Simple H.264의 부호화 시간 및 PSNR
Fig. 5. Encoding time and PSNR plot between H.264 and Simple H.264.

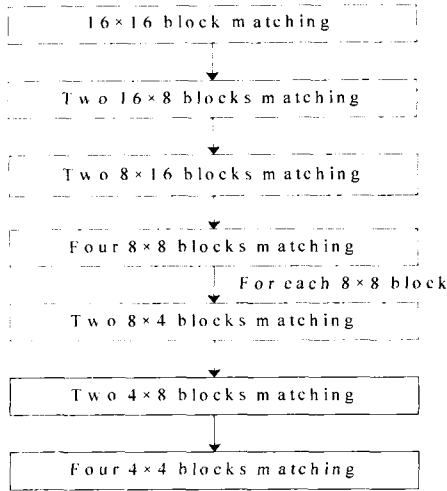


그림 6. 가변 블록 움직임 추정 순서
Fig. 6. The order for the variable block-based motion estimation.

정 후, RD 측면에서 큰 블록모드가 유리하다고 판단되는 매크로 블록들에 대해 P8x8 모드의 움직임 추정 및 $RDcost$ 계산을 skip하는것이다. 빠른 계산을 위해 식 (2)에서 정의된 $Distortion$ 과 비트율 대신 SAD와 $MVcost$ 를 이용하여 P8x8모드에 대한 움직임 추정 및 $RDcost$ 계산의 수행 여부를 판단하였다. 참고로 $MVcost$ 는 움직임 추정 전 이웃 블록들로부터 얻은 예측벡터(Predicted Motion Vector)와 움직임 추정 후 얻은 실제 움직임 벡터 사이의 차분을 UVLC(Universal VLC) 한 값에 의하여 결정된다. 예측벡터와 움직임벡터간의 값의 차이가 크다면 $MVcost$ 는 커지고 반대로 예측벡터와 움직임 벡터 사이의 값이 유사하다면 $MVcost$ 는 작아진다. Inter16x16, Inter16x8, Inter8x16 모드의 $SAD + MVcost$ 는 다음과 같이 나타낼 수 있다.

$$\begin{aligned}
 Inter16 \times 16_SAD + MVcost &= SAD_1 + MVcost_1 \\
 Inter16 \times 8_SAD + MVcost &= SAD_{21} + SAD_{22} + MVcost_{21} + MVcost_{22} \\
 Inter8 \times 16_SAD + MVcost &= SAD_{31} + SAD_{32} + MVcost_{31} + MVcost_{32} \quad (4)
 \end{aligned}$$

식 (4)에서 SAD_1 은 16x16블록의 SAD값, SAD_{21} 은 매크로 블록내의 첫 번째 16x8블록의 SAD값, SAD_{22} 는 두 번째 16x8블록의 SAD값, $MVcost_{21}$, $MVcost_{22}$ 는 각각의 16x8블록들에 대한 $MVcost$ 이고, SAD_{31} , SAD_{32} , $MVcost_{31}$, $MVcost_{32}$ 는 8x16블록의 SAD와 $MVcost$ 를

나타낸다. 일반적으로 $SAD_1 \geq SAD_{21} + SAD_{22}$, $SAD_1 \geq SAD_{31} + SAD_{32}$ 가 성립한다.

$$\Delta SAD = SAD_1 - (SAD_{21} + SAD_{22}) \quad (5)$$

식 (5)와 같이 Inter16x16 모드의 SAD값과 Inter16x8 모드의 SAD값의 차분을 ΔSAD 라 하면 매크로 블록 내의 두 개의 16x8블록이 다른 방향의 움직임을 가졌을 때는 ΔSAD 값이 커지고 유사한 방향의 움직임을 가졌을 때는 ΔSAD 값이 작아지게 된다. 두개의 16x8블록이 완전히 같은 방향의 움직임을 가졌을 때는 ΔSAD 값은 0이 된다. 16x16블록과 8x16블록모드의 SAD값의 차분도 이와 같이 구할 수 있다. 매크로 블록이 하나의 16x16블록으로 움직임 추정을 한 Inter16x16모드와 수평으로 두 개로 나눈 Inter16x8모드와 수직으로 두개로 나눈 Inter8x16모드의 $Inter16 \times 16_SAD + MVcost$, $Inter16 \times 8_SAD + MVcost$, $Inter8 \times 16_SAD + MVcost$ 값을 비교하여 $Inter16 \times 16_SAD + MVcost$ 값이 가장 작다면 매크로 블록을 수직으로 나누었을 때와 수평으로 나누었을 때 모두 하위 블록들의 움직임 벡터가 유사하여 SAD가 줄어드는 ΔSAD 보다 추가적인 $MVcost$ 가 더 크고 매크로 블록 전체가 수직방향 또는 수평방향으로 나누어지지 않고 유사한 방향으로 움직임을 가진다는 것을 뜻한다. 이 경우 더 많은 움직임 벡터를 요구하는 P8x8모드 역시 ΔSAD 보다 $MVcost$ 가 더 클 것으로 예상되므로 많은 계산량을 요구하는 움직임 추정 및 $RDcost$ 계산을 skip할 수 있다. 앞에서 언급했듯이 P8x8모드의 계산량은 다른 모드에 비해 매우 많이 요구되기 때문에 P8x8모드의 움직임 추정 및 윌-왜곡 최적화 과정을 효과적으로 skip 한다면 부호화기의 계산량 및 복잡도를 크게 감소시킬 수 있을 것이다.

2. 공간예측 부호화 과정 생략법

H.264 에서는 가변 블록 크기의 움직임 수행 후, 공간예측 모드 및 SKIP모드에 대한 $RDcost$ 를 비교하여 $RDcost$ 가 최소가 되는 모드를 부호화 모드로 결정한다. 이 과정에서 모든 매크로 블록들에 대한 공간예측 부호화가 수행된다. 그러나 실제 영상을 부호화 할 때 공간예측 부호화는 움직임 추정에 의한 Inter 부호화에 비해 훨씬 많은 비트를 요구하기 때문에 매크로 블록의 부호화 모드가 공간예측 모드로 결정되는 경우는 장면 전환 같은 특별한 경우 외에는 거의 일어나지 않는다. <표 2>는 H.264 표준화 그룹에서 권고하는 실험

조건^[12]을 이용하여 부호화 하였을 때 각 영상의 Predictive프레임에서 공간예측 모드로 부호화 되는 매크로 블록의 비율을 나타내고 있다. <표 2>에서 확인할 수 있듯이 공간예측 부호화 매크로 블록의 비율은 모든 영상에서 아주 작은 비율로 나타나고 있다. 그럼에도 불구하고 모든 매크로 블록에 대해 공간예측 부호화를 수행하는 것은 실제 공간예측 모드로 부호화 되는 매크로 블록의 비율을 감안할 때 많은 계산량의 낭비라 할 수 있다. 이에 본 장에서는 불필요한 공간예측 부호화를 효율적으로 skip할 수 있는 방법을 제안한다. 제안되는 방법을 구현하기 위해서는 먼저 이전 프레임들 및 현재 프레임에서 공간예측 모드로 부호화된 모든 Intra매크로 블록의 RD_{cost} 의 평균인 M 을 구해야 한다. M 의 초기값은 영상의 첫 번째 프레임인 I 프레임의 RD_{cost} 평균값에서 시작하여 공간예측 모드로 부호화 되는 매크로 블록이 발생할 때마다 M 이 갱신된다. 이때 갱신된 M 은 식 (6)과 같이 계산하였다. 식 (6)에서 M' 는 Intra 매크로 블록발생시 RD_{cost} 값,

$$M \leftarrow \frac{1}{n+1}(M'+nM) \quad (6)$$

n 은 현재까지 발생한 Intra 매크로 블록의 갯수이다. 간단한 연산을 통하여 M 을 계속적으로 갱신할 수 있음을 알 수 있다. 가변 블록 크기의 움직임 추정 및 울-왜곡 최적화를 통한 최적 Inter 매크로 블록의 모드를 결정 후, 최적 Inter 매크로 블록의 RD_{cost} 와 M 을 비교하여 최적 Inter매크로 블록의 RD_{cost} 가 더 작다면 움직임 추정이 효율적으로 이루어졌고 공간예측 모드로 부호화 시 최적 Inter모드보다 RD_{cost} 가 커질 확률이 높으므로 공간예측 모드 부호화 과정을 skip하고 매크로 블록의 부호화 모드를 결정한다. 제안된 방법은 P8×8모드의 울-왜곡 최적화시에도 같은 방법으로 사용될 수 있다. P8×8모드에서는 각각의 8×8 블록들에 대해서 Inter8×8, Inter8×4, Inter4×8, Inter4×4 모드 중 최적 모드의 RD_{cost} 와 $M/4$ 을 비교하여 IBLOCK모드의 RD_{cost} 계산을 skip할 수 있다. M 은 고정된 값이 아니라 영상의 공간적 상관관계에 따라 변화하기 때문에 모든 영상에서 효율적으로 적용될 수 있다. 제안되는 방법의 알고리즘을 <그림 7>에서 pseudo code 형태로 나타내었다. <그림 8>은 본 장에서 제안한 가변 블록 크기의 움직임 추정에서의 고속 모드 결정법과 공간예측 부호화 과정 생략법을 이용한 고속모드 결정법의

표 2. 시험 영상의 공간예측 매크로 블록 (Intra 매크로 블록) 비율

Table 2. Ratio of spatial predictive macroblock (intra macroblock) in test sequences.

영 상	QP	Intra MB(%)	영 상	QP	Intra MB(%)
News (QCIF)	28	1.73	Paris (CIF)	28	1.01
	32	1.80		32	1.10
	36	2.03		36	1.18
	40	2.01		40	1.14
Container (QCIF)	28	1.17	Mobile (CIF)	28	0.18
	32	1.24		32	0.17
	36	1.41		36	0.19
	40	1.11		40	0.22
Foreman (QCIF)	28	6.03	Tempete (CIF)	28	2.21
	32	5.90		32	2.09
	36	5.61		36	1.92
	40	5.31		40	1.66
Silent (QCIF)	28	2.58	Average	28	2.13
	32	2.63		32	2.13
	36	2.63		36	2.14
	40	2.64		40	2.01

Inter_mode RDs(); // Calculate Inter mode Rate Distortions

Best_Inter_mode RDcost = Best_Inter_mode_decision();

If (Best_Inter_mode RDcost < M)

SKIP Intra mode RD calculation

else

Intra_mode RD(); // Calculate Intra mode Rate Distortion

Encoding_mode = Encoding_mode_Decision(); // Decide Intra/Inter mode

If (Encoding_mode == Intra)

Update M value by eq.(6);

그림 7. 공간예측 부호화 생략 알고리즘

Fig. 7. Skip algorithm of spatial predictive coding.

흐름도를 나타내고 있다. 매크로 블록 단위에서 Inter16×16, Inter16×8, Inter8×16 세 모드의 $SAD + MV_{cost}$ 를 비교하여 $Inter16 \times 16_{SAD} + MV_{cost}$ 가 최소 이면 블록이 나누어지기보다는 큰 블록모드를 유지하

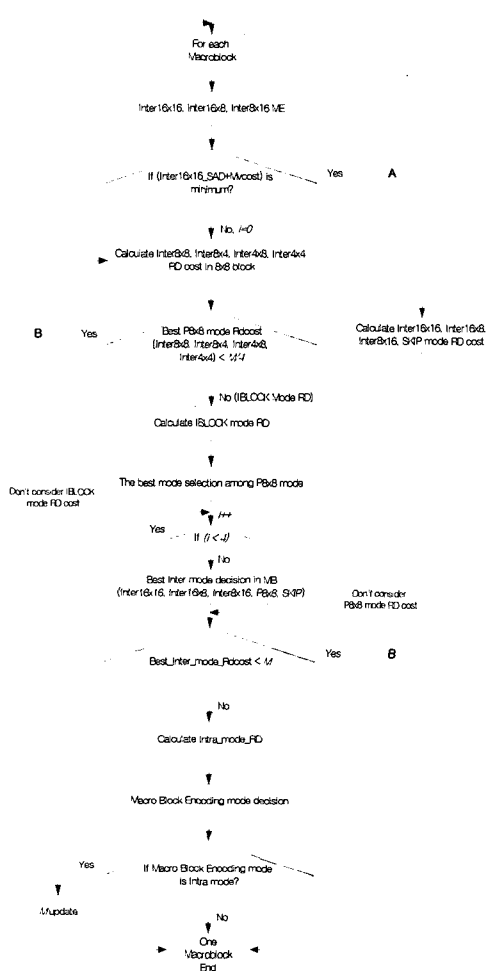


그림 8. 제안된 고속모드 결정법 흐름도
Fig. 8. Flow chart of proposed fast mode decision method.

는 것이 유리하다고 판단되므로 A 방향으로 빠져서 많은 계산량을 요구하는 P8×8 모드의 ME 및 RDcost 계산을 생략한다. P8×8모드의 움직임 추정 및 RDcost 계산 여부를 판단한 후, Inter16×16, Inter16×8, Inter8×16, SKIP, P8×8모드 중에서 최적의 Inter모드를 선택한다. 최적 Inter모드의 RDcost를 M과 비교하여 최적 Inter모드의 RDcost가 작다면 움직임 추정이 효율적으로 이루어졌음을 의미하므로 B 방향으로 빠져 매크로 블록 단위의 공간예측 부호화를 생략한다. 반대의 경우는 공간예측 부호화 모드의 RDcost를 계산한 후, 최적 Inter모드의 RDcost와 비교하여 매크로 블록의 부호화 모드를 결정한다. 매크로 블록의 부호화 모드가 공간예측모드로 결정된다면 M을 갱신한 후 종료한다. 공간예측

부호화를 생략하는 방법은 P8×8모드에서도 적용할 수 있다. Inter8×8, Inter8×4, Inter4×8, Inter4×4모드 중 최적 모드의 RDcost와 M/4을 비교하여 최적 모드의 RDcost가 작다면 IBLOCK 모드의 RDcost계산을 생략할 수 있다. 흐름도에서 A는 가변 블록 크기의 움직임 추정에서의 빠른 모드 결정법을 통하여 B는 공간예측 부호화 과정 생략법을 통하여 RDcost계산을 생략하는 부분이다.

IV. 실험 결과

본 논문에서 제안된 고속모드 결정법의 성능을 측정하기 위한 실험은 H.264 표준화 그룹에서 권고하는 실험 조건을 따라 <표 3>과 같이 수행 되었다. 다만 Tempete영상은 실험시 226장의 프레임만 사용하였다. JM42(Joint Model 42)^[13] 코덱을 이용하여 제안된 고속모드 결정법의 성능을 실험하였다. <표 4>는 제안된 고속모드 결정법과 JM42의 성능을 비교한다. <표 4>에서 ΔBits와 ΔPSNR은 H.264와 제안된 고속모드 결정법간의 비트율과 PSNR의 차이로서 식 (7), 식 (8)과 같이 계산된다. H.264 표준화 그룹에서는 비트율의 차이는 %로, PSNR의 차이는 차분값으로 비교하는 것을 권고한다^[12]. 따라서 본 논문에서는 ΔBits와 ΔPSNR을 이용하여 실험결과를 보인다. 참고로 ΔBits와 ΔPSNR의 (-)부호는 성능의 향상을 뜻한다. A는 공간예측 부호화 과정에서의 RD 계산 감소량을, B는 움직임 추정에 사용되는 가변 블록 모드에서의 RD 계산 감소량을, Total은 전체 부호화 과정에서의 RD 계산 감소량을 나타낸다. 줄어든 계산감소량은 식 (9)와 같이 구하였다. 실험결과, Intra 부호화시 사용되는 공간예측 부호화 과정에서 평균 94% 이상의 계산이 감소하며 Inter 부호화시 사용되는 가변 블록 크기 움직임 추정에서 31~39% 이상의 계산이 감소함을 확인 할 수 있다.

$$\Delta Bits = \frac{\text{제안된 방법의 Bits} - \text{JM42의 Bits}}{\text{JM42의 Bits}} \times 100(\%) \quad (7)$$

$$\Delta PSNR = \text{JM 42의 PSNR} - \text{제안된 방법의 PSNR} \quad (8)$$

계산감소량 =

$$\frac{\text{JM42의 RDcost계산 수행횟수} - \text{제안된 방법의 RDcost계산 수행횟수}}{\text{JM42의 RDcost계산 수행횟수}} \times 100(\%) \quad (9)$$

공간예측 모드와 가변블록 모드 및 SKIP 모드까지 포함된 전체 RD 계산 수행횟수도 평균 75% 이상 감소함을 확인 할 수 있다. 이처럼 많은 계산량의 감소에 비해 비트율은 평균 0.69%, PSNR은 평균 0.055의 손실이 있음을 알 수 있다. 제안된 고속모드 결정법에 의한 계산감소량을 고려할 때 이는 아주 미미한 수준이라 할 수 있다. <그림 9>에 <표 3>에서 주어진 여러 QCIF, CIF 해상도를 갖는 시험용 sequence들에 대하여 제안된 고속 모드 결정법(FastMode), JM42, 2장에서 언급한 Simple H.264를 적용했을 때 각각의 sequence에서 3가지 방법에 대한 bitrates 대 PSNR의 성능을 비교하였다. 제안된 고속모드 결정법은 H.264와 거의 같은 곡선을 갖는 것을 확인할 수 있다. 이는 고속모드 결정법의 부호화 효율이 H.264와 거의 같다는 것을 의

표 3. H.264 권고 실험 조건

Table 3. Recommended simulation conditions for H.264.

	News (QCIF)	Container (QCIF)	Foreman (QCIF)	Silent (QCIF)	Paris (CIF)	Mobile (CIF)	Tempete (CIF)
Total frame	300	300	300	300	300	300	260
Frame skip	2	2	2	1	1	0	0
QP	28, 32, 36, 40						
Coding Options	가변 크기 블록 움직임 추정, 울-왜곡 최적화, 하다마드 변환, B프레임 사용안함(IPPPP...), CAVLC, 어러틀 사용안함						

표 4. 고속모드 결정법과 H.264의 성능비교

Table 4. Performance comparison of proposed fast mode decision with H.264.

	QP	ΔBts (%)	ΔPSNR (dB)	A ^(%)	B ^(%)	Total (%)		QP	ΔBts (%)	ΔPSNR (dB)	A ^(%)	B ^(%)	Total (%)
News	28	1.20	0.03	94.84	36.67	75.83	Paris	28	0.40	0.03	94.98	35.97	75.96
	32	1.35	0.04	94.30	37.38	76.05		32	0.40	0.05	95.24	36.51	76.85
	36	1.18	0.05	94.78	38.33	76.68		36	0.16	0.03	95.21	40.46	77.30
	40	1.65	0.03	94.48	42.71	77.17		40	-0.12	0.07	95.30	41.72	77.63
Container	28	0.99	0.05	94.31	38.70	75.94	Mobile	28	0.07	0.06	96.98	27.87	75.21
	32	0.91	0.04	93.16	40.81	75.58		32	0.15	0.05	97.95	28.12	75.96
	36	0.73	0.03	93.61	43.31	74.63		36	0.00	0.05	98.11	29.74	76.30
	40	0.87	0.13	94.83	45.67	74.72		40	0.00	0.08	97.63	33.95	77.27
Foreman	28	1.30	0.05	94.11	39.34	76.23	Tempete	28	0.36	0.04	93.12	25.32	72.06
	32	0.80	0.08	94.93	38.89	76.01		32	0.32	0.04	94.49	28.04	75.53
	36	1.11	0.07	94.20	38.70	76.48		36	0.32	0.05	94.80	32.04	74.79
	40	0.53	0.16	94.30	33.45	72.51		40	0.83	0.03	94.74	37.01	76.05
Silent	28	1.48	0.05	94.60	33.24	76.12	Average	28	0.84	0.04	94.32	31.02	74.06
	32	0.66	0.09	94.63	32.88	75.96		32	0.60	0.04	94.61	33.04	74.72
	36	1.36	0.05	94.21	33.38	74.40		36	0.68	0.06	94.79	31.14	75.41
	40	0.97	0.09	94.09	43.92	76.01		40	0.63	0.08	94.18	30.65	76.30

A: 공간예측 부호화 과정에서 RD 계산 감소량

B: 움직임 추정에서 RD 계산 감소량

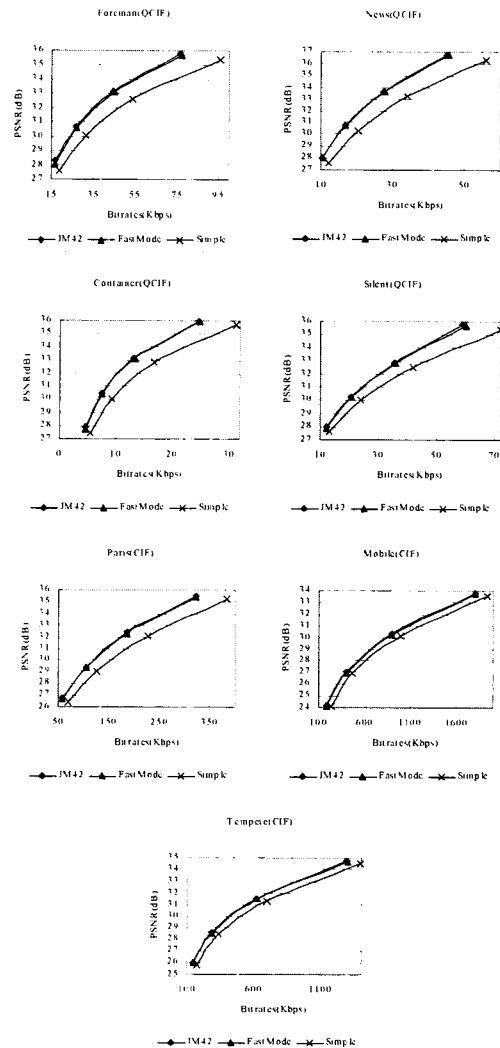


그림 9. 고속 모드 결정법과 H.264 및 Simple의 PSNR 그래프

Fig. 9. PSNR graph of proposed fast mode decision and H.264 and Simple H.264.

미한다. <그림 9>와 <표 4>로부터 고속모드 결정법으로 부호화시 H.264의 부호화 효율을 유지하면서도 많은 계산량의 이득을 얻을 수 있음을 확인 할 수 있다.

V. 결론

본 논문에서는 H.264의 부호화기의 복잡도에서 가장 많은 부분을 차지하는 가변 블록 크기 움직임 추정 및 공간예측 부호화를 생략하여 울-왜곡 최적화를 통한 부호화 모드 결정의 복잡도를 감소시켜 주는 고속 모드 결정법을 제안하였다. 제안된 방법은 울-왜곡 최적

화시 $RDcost$ 계산횟수를 평균 75% 이상 감소하면서도 부호화 효율을 나타내는 두 가지 척도인 비트율 및 PSNR의 손실은 아주 적은 것을 확인 할 수 있다. 특히 공간예측 부호화를 생략하는 방법은 공간예측 모드의 계산횟수를 94% 이상 줄여주어 부호화기의 복잡도 감소에 크게 기여할 것으로 예상된다. 그에 반해 이전에 공간예측모드로 부호화된 매크로 블록들의 $RDcost$ 의 평균인 M 은 매우 간단한 연산으로 구할수 있으며 새롭게 공간예측 모드로 부호화 되는 매크로 블록이 발생 할 때도 쉽게 갱신 할 수 있다. 본 논문에서 제안한 방법들은 고속 움직임 추정 알고리즘과 결합하여 고속 부호화기의 구현시 유용하게 사용될 것으로 예상된다.

참 고 문 헌

[1] ITU Telecom. Standardization Sector, "Video Codec Test Model Near-Term, Version 10 (TMN10) Draft 1," H.263 Ad Hoc Group, April 1998.

[2] "Information Technology - Coding of Audio-Visual Objects Part2: Visual Amendment 1: Visual Extensions", ISO/IEC JTC1/SC29/WG11 N3056, Dec. 1999.

[3] Thomas Wiegand, Joint Final Committee Draft (JFCD) of Joint Video specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC), JVT-G050, March 2003.

[4] Antonio Ortega and Kannan Ramchandran, "Rate-Distortion Methods for Image and Video Compression", IEEE SIGNAL PROCESSING MAGAZINE, pp23-50, November 1998.

[5] Gary J.Sullivan and Thomas Wiegand, "Rate-Distortion Optimization for Video Compression", IEEE SIGNAL PROCESSING MAGAZINE, pp 74-90, November 1998.

[6] P. Topiwala, G. Sullivan, A. Joch, F. Kossentini, "Overview and Performance Evaluation of the Draft ITU-T H.26L Video Coding Standard", Proc. SPIE, Appl. Dig. Im. Proc, Aug.2001.

[7] ITU-T SG16 Q6, "Performance Evaluation of H.26L, TML 8 vs H.263++ and MPEG-4", VCEG-N18, Sep.2001.

[8] M.Zhou, "Evaluation and Simplification of H.26L Baseline Coding Tools", JVT-B030, Jan, 2002.

[9] K. Takagi, "Lagrange Multiplier and RD-characteristics", JVT-C084, May, 2002.

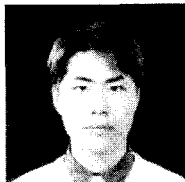
[10] G. Bjontegaard and K. Lillevold "Context-adaptive VLC(CVLC) coding of coefficients", JVT-C028, May, 2002.

[11] James Au, "Complexity Reduction for CAVLC", JVT-D034, July, 2002.

[12] G. Sullivan and G. Bjontegaard, "Recommended simulation common conditions for H.26L coding efficiency experiments on low-resolution progressive-scan source material", ITU-T Q.6/16, Doc. #VCEG-N81, Sep. 2001.

[13] <http://standard.pictel.com/ftp/videosite/h261/jm42.zip>

저 자 소 개



韓 基 勳(學生會員)
2003년 2월 : 세종대학교 컴퓨터공학과 졸업. 2003년 3월~현재 : 세종대학교 인터넷공학과 석사과정.
<주관심분야 : 영상처리, 멀티미디어 응용, 컴퓨터비전>



李 英 烈(正會員)
1985년 2월 : 서강대학교 전자공학과 졸업. 1987년 2월 : 서강대학교 전자공학과 석사. 1999년 2월 : 한국 과학 기술원 전기 및 전자공학과 박사. 2001년 8월 : 삼성전자 중앙연구소 DMS Lab. 수석연구원. 2001년 9월~ 현재 : 세종대학교 인터넷학과 조교수.
<주관심분야 : 영상처리(압축,복원), 영상전송, 멀티미디어 시스템, 비디오 Transcoding>