

# H.264에서의 고속 매크로블록 모드 선택 알고리즘

정회원 김 동 형\*, 정 제 창\*\*

## A Fast MB Mode Selection Algorithm in the H.264 Standard

Donghyung Kim\*, Jechang Jeong\*\* *Regular Members*

### 요 약

H.264 부호화 표준은 부호화 효율을 높이기 위해 1/4 화소 단위의 움직임 추정, 다중 참조 프레임, 인트라 예측, 루프 필터, 다양한 블록 크기의 지원 등과 같은 새로운 부호화 도구들을 사용한다. 이를 통해 이전의 비디오 부호화 표준들에 비해 율-왜곡(率-歪曲) 관점에서 높은 성능을 보이지만 그로 인해 부호기의 복잡도는 상당히 증가한다. 본 논문은 부호기 복잡도의 증가를 초래하는 주요 부호화 도구들 중 매크로블록 모드 선택의 복잡도 감소에 주안(主眼)점을 두며, 이에 대한 복잡도 감소 알고리즘을 제시한다. 선택가능한 모든 매크로블록 모드들 중에서 8×8 모드와 인트라 4×4 모드는 다른 모드들에 비해 상대적으로 높은 복잡도를 가진다. 제안하는 알고리즘은 낮은 복잡도를 가지는 매크로블록 모드의 비용(Cost) 정보를 이용하여 8×8 및 인트라 4×4의 비용을 추정함으로써 두 모드에 대한 복잡도를 감소시키고자 한다. 제안한 알고리즘을 적용한 실험은 전체 부호화 시간을 최대 54.6%까지 감소시키는데 반해 PSNR은 단지 평균 0.012dB 감쇠(減衰)하는 것으로 나타난다.

Key Words : H.264 video coding; Encoder complexity; Mode selection; Intra prediction.

### ABSTRACT

For the improvement of coding efficiency, the H.264 standard uses new coding tools such as VBS, 1/4-pel accurate ME, multiple references, intra prediction, loop filter, etc. Using these coding tools, H.264 has achieved significant improvements from rate-distortion point of view compared to existing standards. However, the encoder complexity is greatly increased due to these coding tools. We focus on the complexity reduction method of MB mode selection. Among all modes which can be selected, 8×8 and intra 4×4 mode have higher complexity than the others. So we propose the methods for reduction of the 8×8 and intra 4×4 mode complexity by using information of other modes with relatively low complexity. Simulation results show that the proposed methods save up to 54.6% of total encoding time while keeping the average decrease about 0.012dB in PSNR.

### 1. 서 론

H.264/AVC 표준은 ITU-T의 VCEG (Video Coding Experts Group)과 ISO/IEC의 MPEG (Moving Picture Expert Group)의 공동 작업으로 제정되었다<sup>[1]</sup>. H.264/AVC 비디오 부호화 방식은 이전의 비디오 부호화 표준들(MPEG2, MPEG4 Part2, H.263)에는 없던 새로운 부호화 도구들이 추가되었으

며, 이러한 도구들은 H.264 부호화 표준이 이전의 비디오 부호화 표준과 비교하여 보다 높은 압축효율을 갖게 하였다. 새롭게 추가된 부호화 도구들을 간단히 살펴보면 다음과 같다<sup>[2]</sup>.

정수 변환 및 양자화 : H.264에서는 4×4 단위의 정수 변환을 수행하며, 부호기와 복호기 사이의 부정합(Mismatch) 문제가 발생하지 않는다. 이는 8×8 단위의 이산여현변환(Discrete Cosine Transform)을 사

\* 한양대학교 전자통신전파공학과 영상처리 및 신호처리 연구실 (kimdh@ece.hanyang.ac.kr)

\*\* 한양대학교 전자전기컴퓨터 공학부 (jjeong@ece.hanyang.ac.kr)

논문번호 : KICS2004-11-272, 접수일자 : 2004년 11월 10일

※ 본 논문은 정보통신부의 출연금으로 수행한 IT SoC 핵심설계인력양성사업의 수행결과입니다.

용함으로써 부호기와 복호기 사이에서의 부정합 문제를 가지고 있던 기존의 여러 비디오 부호화 표준들과 비교된다. 또한 매크로블록이 인트라 16×16 모드로 선택된 경우 변환수행 단위인 4×4 변환블록의 DC값들은 다시 한번 하다마드(Hadamard Transform) 변환을 수행하며, 이러한 계층적 변환(Hierarchical Transform)을 통해서 보다 높은 압축효율을 나타낸다. 또한 H.264에서의 양자화 파라미터는 양자화 데이터를 가리키는 하나의 인덱스(Index)로 사용되며 0~51까지 52단계의 값을 가진다. 양자화 스텝 크기(Quantization Step Size)는 양자화 파라미터가 여섯 단계 증가함에 따라 두 배로 증가한다.

**엔트로피 부호화** : H.264에서는 프로파일에 따라 두 가지의 엔트로피 부호화 방법을 사용한다. 첫 번째 엔트로피 부호화 기법은 변환 계수를 제외한 대부분의 신택스(Syntax)를 데이터의 특성에 따라 하나의 코드워드(Codeword) 테이블로 변환 후 EGC(Exponential Golomb Code)로 부호화 하고, 변환계수는 이미 부호화 된 주변 블록들의 특성을 이용하는 CAVLC(Context Adaptive Variable Length Coding)로 부호화 하는 방법이다. 두 번째의 엔트로피 부호화 기법은 기존의 산술 부호화 방식에 컨텍스트 모델링(Context Modeling)을 추가함으로써 추정된 심볼(Symbol) 확률 분포 모델들을 적응적으로 선택하는 방식으로 EGC와 CAVLC를 이용하는 엔트로피 부호화 방식에 비해 높은 복잡도를 가지지만 5~15%의 비트율 감소효과가 있다.

**루프 필터** : 블록 기반의 움직임 추정 및 변환 그리고 양자화 기법은 부호기 설계에 많은 이점을 가져 오지만 블록화 현상(Blocking Artifact)을 수반한다. H.264는 이러한 블록화 현상의 효율적인 개선을 위해 부호화 및 복호화 과정 내(內)에 루프 필터를 사용하여 블록화 현상을 제거한다. 이는 부호화 및 복호화 과정과 독립적인 후처리(Post-Processing)의 수행을 통해서 블록화 현상을 제거하는 기존의 부호화 표준들과 비교된다.

**인트라 예측 부호화** : H.264 부호화 기법은 공간영역에서의 인트라 예측 부호화 과정을 포함한다. 공간 내의 예측 부호화 방식은 휘도(Luminance) 성분의 경우 16×16크기 및 4×4크기로 수행되며, 각각 4가지와 9가지의 공간내의 방향성을 고려한 예측모드를 가지고 있고, 색차(Chrominance) 성분은 16×16 크기의 휘도성분과 같은 4가지의 예측모드를 가지고 있다. 휘도 성분에 대한 공간영역에서의 16×16 크기와 4×4 크기의 인트라 예측은 영상의 평탄한 영역뿐만 아니

라 세밀한 영역에서도 높은 압축 효율을 갖게 한다. 공간 영역(Spatial Domain)에서의 인트라 예측 부호화 방식은 변환 영역(Transform Domain)에서 휘도 및 색차 신호에 대한 인트라 예측 부호화를 수행하는 MPEG-4의 표준과 비교된다.

**다중 참조 영상 및 1/4 화소 단위의 움직임 추정** : H.264에서는 참조 영상으로서 이전의 재구성된 여러 장의 프레임을 사용한다. 이러한 다중 참조 영상의 사용은 단일 참조 영상만을 사용하는 경우에 비해 보다 높은 압축 효율을 가지게 하며, 특히 반복적인 영상이나 다른 객체에 의해 가려져 있다가 나타난 영역의 부호화 경우에 더욱 두드러진 성능을 보인다. 또한 H.264에서는 보다 정확한 움직임 추정을 위해 보간(Interpolation) 기법을 이용하여 1/4 화소 단위로 움직임을 추정한다. 이때 1/2 화소의 보간 방식으로는 6탭 필터를 사용하며, 1/4 화소 단위의 보간 방법은 쌍선형 필터(Bilinear Filter)를 사용한다.

**다양한 블록크기의 움직임 추정** : H.264에서는 블록의 크기를 다양화하여 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4의 크기로 움직임을 추정한다. 이는 기존에 16×16 및 8×8 크기로만 움직임을 보상하던 H.263 및 MPEG-4 기술과 대비된다. 다양한 크기의 블록 단위, 특히 8×8 부-매크로블록(Sub Macroblock) 이하의 크기인 8×4, 4×8, 4×4 블록 크기의 움직임 추정은 보다 세밀한 영상 영역에서도 높은 압축 효율을 가질 수 있게 한다.

그 밖에도 적응적 프레임/필드 부호화(Adaptive Frame/Field Coding) 및 NAL(Network Abstraction Layer) 등의 부호화 도구들이 추가되었다. 이렇게 새롭게 추가된 부호화 도구들은 H.264 부호화 표준이 이전의 여러 동영상 부호화 표준들에 비해 보다 높은 압축 효율을 가질 수 있게 한다. 다섯 장의 참조 프레임의 사용과 32 화소의 움직임 추정 영역을 사용하여 메인 프로파일로 부호화 한 경우, H.264 표준은 MPEG-2와 비교하여 약 48~78%까지의 비트율 감소 효과가 있다고 알려져 있다<sup>[3]</sup>. 하지만 비트율 감소 효과와 더불어 부호기의 복잡도는 상당히 높아진다. 특히 새로이 추가된 부호화 도구들 중 다중 참조 영상의 사용 및 1/4화소 단위의 움직임 추정과 다양한 블록 크기에 대한 율-왜곡(Rate-Distortion) 측면에서의 비용함수 계산 및 비교는 부호기 복잡도 증가의 주된 요인이 된다.

실시간 H.264 부호화의 실현을 위해 부호기의 복잡도 감소에 대한 연구는 H.264 부호화 표준이 제정된 이후 다각적으로 연구되어 왔다. Peng 등은 부호

기의 복잡도 감소를 위하여 두 가지 알고리즘을 제안하였다<sup>4)</sup>. 첫 번째는 효율적인 매크로블록의 모드선택을 위한 알고리즘으로서 P 프레임 내 하나의 매크로블록이 가질 수 있는 모드인 SKIP, 16×16, 16×8, 8×16과 부-매크로블록이 가질 수 있는 모드인 8×8, 8×4, 4×8, 4×4 그리고 16×16 크기 및 4×4 크기의 공간 내 예측을 위한 인트라 모드 중에서 인트라 모드를 제외한 SKIP~4×4까지의 매크로블록 모드의 고속 선택을 위해 SKIP, 16×16, 8×8, 4×4의 매크로블록 모드에 대한 비용(Cost)을 이용하여 잔여 매크로블록 모드의 계산여부를 결정함으로써 부호기의 복잡도를 감소시킨다. 또한 매크로블록 모드 중 SKIP 모드에 가중치를 줌으로써 매크로블록 모드의 잘못된 추정을 통한 비트율의 증가를 최소화 하였다. 인트라 모드의 경우는 앞서 계산한 모드에서의 비용이 임계값 이상일 경우만 추가적으로 계산함으로써 인트라 모드의 계산량 또한 감소시켰다. 또한 Peng은 정수화 소 단위의 움직임 추정을 통해 얻은 정보를 1/2화소 및 1/4화소 단위의 움직임 추정에 사용함으로써 고속 움직임 추정을 위한 알고리즘을 제안하여 부호기 복잡도를 감소하였다. Huang 등은 다중 참조 프레임을 사용하는 H.264의 부호화 과정의 복잡도를 줄이기 위해 이전 한 프레임에서의 움직임 추정 및 인트라 예측을 통해 얻어진 정보를 이용하여 잔여 참조 프레임의 사용여부를 결정하는 알고리즘을 제안하여 10%~67%까지 움직임 추정시간의 감소 효과를 가져왔다<sup>5)</sup>. Ahmad등은 수정된 3DRS(3D Recursive Search)를 사용하여 매크로블록 모드의 고속 선택 기법을 제안하였다<sup>6)</sup>. 이웃한 매크로블록과 이전 프레임 내의 같은 위치에 있는 매크로블록의 모드 및 각각의 움직임 벡터의 비용(Cost)을 이용하여 현재 매크로블록의 모드를 추정함으로써 전체 부호화 시간을 약 30% 감소시켰다. 하지만 비트율은 최대 15%까지 증가하였다.

본 논문은 P 프레임 내의 매크로블록 모드 선택과정에 있어서 상대적으로 낮은 복잡도를 갖는 16×16, 16×8 및 8×16 크기의 움직임 추정으로 발생하는 움직임 벡터 비용(MVcost) 및 참조 프레임 비용(REFcost)을 사용하여 부-매크로블록의 모드를 추정한다. 선택 가능한 부-매크로블록의 모드를 추정된 모드로 제한함으로써 부-매크로블록의 모드를 결정하는데 사용되는 윌-왜곡 비용함수(REFcost)의 계산회수를 감소시키고, 그에 따른 부호기의 복잡도를 감소시키는 알고리즘을 제안한다. 또한 P 프레임 내의 매크로블록 모드 중 인트라 4×4 모드는 윌-왜곡 비용 측

면에서 상당한 상관성이 있는 인트라 16×16 모드의 윌-왜곡 비용으로 예측함으로써 이에 대한 계산량을 줄일 수 있는 알고리즘을 제안한다.

논문의 구성은 2절에서 참조 소프트웨어(JM 8.4)에서의 매크로블록 모드 선택 알고리즘에 대해서 기술하고 3절에서는 논문에서 제안하는 부-매크로블록 복잡도 감소 알고리즘 및 인트라 4×4 모드의 계산량 감소 알고리즘을 기술한다. 4절에서는 8개의 QCIF 크기의 테스트 시퀀스에 대해서 제안하는 알고리즘을 적용함으로써 알고리즘의 효율성을 보이고 마지막 절에서 결론을 맺는다.

## II. H.264/AVC내에서 매크로블록 모드의 선택 알고리즘

H.264/AVC 부호화 과정에서 I 프레임 내의 매크로블록은 인트라 16×16 또는 인트라 4×4 모드로 선택될 수 있다. 반면 P 프레임 내의 매크로블록은 좀더 다양한 모드를 가지는데, 선택 가능한 모드로는 16×16, 16×8, 8×16, 8×8 모드가 있으며, 8×8 크기를 갖는 부-매크로블록 모드는 다시 8×4, 4×8, 4×4의 블록 크기로 나눌 수 있다. 또한 I 프레임 내의 매크로블록 모드인 인트라 16×16과 인트라 4×4 모드 선택 대상이며 추가적으로 SKIP 모드를 가질 수 있다. 따라서 P 프레임 내의 매크로블록은 SKIP, 16×16, 16×8, 8×16, 8×8 모드 및 인트라 16×16 그리고 인트라 4×4 모드 중에서 선택된다. 앞서 기술(記述)한 바와 같이 8×8 크기를 갖는 부-매크로블록은 8×4, 4×8, 4×4의 블록 크기로 나뉘어 질수 있기 때문에 P 프레임 내의 매크로블록은 1~16개까지의 움직임 벡터를 가질 수 있다.

윌-왜곡 최적화 기법이 사용되는 경우, 매크로블록 모드의 선택은 모든 선택 가능한 모드의 비용함수를 계산하여 최소의 비용을 가지는 모드를 선택하는데 이때 사용되는 비용함수가 윌-왜곡 비용 함수이다. H.264는 이러한 다양한 매크로블록 모드와 윌-왜곡 비용함수의 사용으로 보다 높은 압축 효율을 나타내지만, 이로 인한 복잡도의 증가는 부호기의 몫으로 남는다.

### 1. 매크로블록 모드의 정의

P 프레임 내의 매크로블록 모드들 중에서 16×16, 16×8, 8×16, 8×8, SKIP 모드는 시(時)방향 예측을 위한 모드이며 각각을 Mode1, Mode2, Mode3, Mode8, SKIP 모드로 정의한다. 그리고 Mode8 내의

부-매크로블록이 선택 가능한 8×8, 8×4, 4×8, 4×4 모드는 각각 Mode4, Mode5, Mode6, Mode7로 정의한다. 또한 인트라 16×16 및 인트라 4×4 모드는 공간영역에서의 예측을 위한 모드로서 I16MB, I4MB 모드로 정의한다.

1.1 시방향 예측을 위한 매크로블록 모드

시방향 예측을 위한 매크로블록 모드에는 Mode1, Mode2, Mode3, Mode8, SKIP이 있으며 부-매크로블록 모드에는 Mode4, Mode5, Mode6, Mode7이 있다. 각각의 블록 모양은 그림 1과 같다.

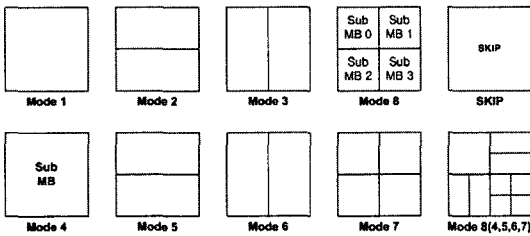


그림 1. 시방향 예측을 위한 매크로블록 모드 및 부-매크로블록 모드

Fig. 1. MB and subMB modes for temporal prediction

그림 1에서 Mode1, Mode2, Mode3, Mode, SKIP은 16×16크기의 매크로블록을 나타내고, Mode8의 경우 다시 8×8크기의 부-매크로블록으로 나뉘며, 각각의 부-매크로블록 모드는 Mode4~Mode7 중에서 선택될 수 있다. 이때 하나의 매크로블록 내의 4개의 부-매크로블록 모드는 독립적으로 선택된다.

$$\text{Mode8(SubMB0, SubMB1, SubMB2, SubMB3)} \\ = \text{Mode8}(4|5|6|7, 4|5|6|7, 4|5|6|7, 4|5|6|7) \quad (1)$$

식 (1)은 4개의 부-매크로블록 모드를 갖는 Mode8의 표현식이며, (4|5|6|7)의 표현은 각각의 부-매크로블록의 모드가 Mode4~Mode7 중에서 선택 가능함을 나타낸다. 그림 1의 가장 오른쪽 하단의 그림은 Mode8내의 각 부-매크로블록에 각각 Mode4, Mode5, Mode6, Mode7이 선택된 경우의 매크로블록 모양을 나타내며, 이에 대한 수식적 표현은 식 (2)와 같이 정의한다.

$$\text{Mode8(SubMB0, SubMB1, SubMB2, SubMB3)} \\ = \text{Mode8}(4, 5, 6, 7) \quad (2)$$

Mode1의 경우 하나의 움직임 벡터만을 가지며 16×16 크기의 블록 단위로 움직임을 추정하고, 영상

의 평탄한 영역 및 배경 영역에서 효율적인 모드이다. Mode2는 두개의 16×8크기의 블록단위로 움직임을 추정하며, 이 경우 두개의 움직임 벡터를 갖는다. 이는 매크로블록 내에 가로방향의 객체가 걸쳐 있는 경우 효율적인 모드이다. Mode3은 두개의 8×16 크기의 블록단위로 움직임을 추정하며, Mode2의 경우와 동일하게 두개의 움직임 벡터를 갖는다. 이는 매크로블록 내에 객체가 세로 방향으로 걸쳐 있는 경우 효율적인 모드이다. Mode8의 움직임 벡터의 개수는 부-매크로블록의 선택 모드에 따라 달라지며 4~16개의 움직임 벡터를 가질 수 있고 세밀한 영상 영역에서 효율적인 모드이다. SKIP 모드의 경우 매크로블록의 움직임 벡터가 예측된 움직임 벡터와 동일하고 화소의 예측오차가 충분히 작은 경우 선택될 수 있으며, 매크로블록 모드가 SKIP이 선택된 경우 부호기는 매크로블록의 모드만을 전송한다.

1.2 공간영역에서의 예측을 위한 매크로블록 모드

공간영역에서의 예측을 위한 매크로블록 모드에는 I16MB, I4MB가 있으며 각각의 블록 모양은 그림 2와 같다.

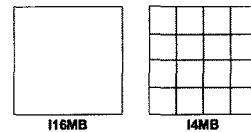


그림 2. 공간영역에서의 예측을 위한 I16MB 및 I4MB 모드 세분화

Fig. 2. Block partitions of I16MB and I4MB modes for spatial prediction

그림 2에서 I16MB 모드의 경우 16×16 크기의 매크로블록을 4가지의 공간 내 예측 방향으로 예측을 수행하며, I4MB의 경우 4×4 크기의 단위로 9가지의 공간 내 예측 방향으로 예측을 수행함으로써 예측 오차를 산출한다. 그림 3과 그림 4는 I16MB 및 I4MB의 공간 내 예측 방향을 도시한다.

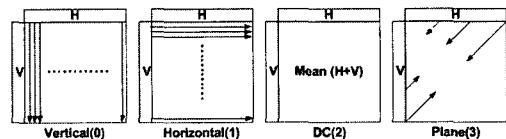


그림 3. I16MB모드에서의 4가지 공간적 예측 모드

Fig. 3. Four prediction modes of I16MB

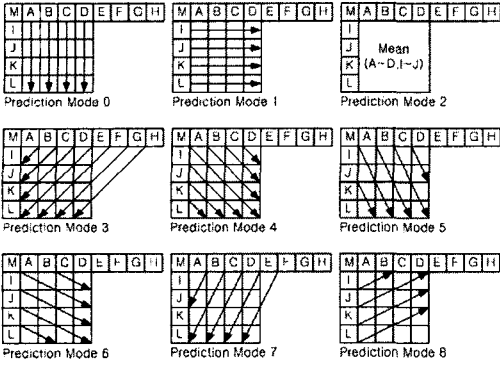


그림 4. I4MB 모드에서의 9가지 공간적 예측 모드  
Fig. 4. Nine prediction modes of I4MB

I16MB 및 I4MB의 예측 방향의 개수 및 예측을 수행하는 블록의 크기를 고려하면, I4MB의 복잡도가 I16MB에 비해 상대적으로 높다는 것을 유추 할 수 있다.

2. JM(Joint Model)에서의 매크로블록 모드 선택

앞서 기술한 바와 같이 P 프레임 내의 매크로블록은 시방향의 예측을 위한 5가지의 매크로블록 모드(SKIP, Mode1, Mode2, Mode3, Mode8) 및 공간 내 예측을 위한 2가지의 매크로블록 모드(I16MB, I4MB) 중 하나로 선택된다. 참조 소프트웨어는 매크로블록의 모드를 선택하기 위해 움직임벡터 비용함수(MVcost), 참조 프레임 비용함수(REFcost) 그리고 율-왜곡 비용함수(RDcost)를 사용한다.

$$MVcost = WeightedCost(f, mvbits[(cx < s) - px] + mvbits[(cy > s) - py])$$

where,  $f = \lambda$  factor

움직임 벡터 비용함수는 식 (3)과 같다. 여기에서,  $cx$  및  $cy$ 는 현재 매크로블록의 움직임 벡터를 나타내고,  $s$ 는 움직임 벡터의 추이(推移) 양을 나타내며,  $px$  및  $py$ 는 예측된 움직임 벡터를 나타낸다.  $WeightedCost$  함수는  $f$  값을 사용하여 움직임 벡터 예측 오차의 부호화에 사용되는 비트량을 인수로 받아 비용 값을 반환한다.

$$REFcost = WeightedCost(f, refbits(ref))$$

where,  $f$ :  $\lambda$  factor

참조 프레임 비용 함수는 식 (4)와 같다. 여기에서  $ref$ 는 선택된 참조 프레임을 가리키며,  $WeightedCost$

함수는  $f$  값을 사용하여 선택된 참조프레임 값을 부호화 하는데 필요한 비트량을 인수로 받아 비용 값을 반환한다. 일반적으로 움직임 벡터 비용에 비하여 매우 적은 값을 갖는다.

$$RDCost = Distortion + \lambda \times Rate \quad (5)$$

율-왜곡 비용 함수는 식 (5)와 같다. 율-왜곡 최적화 기법이 사용되는 경우 율-왜곡 비용 함수는 Mode8 내의 부-매크로블록 모드의 선택 및 전체 매크로블록 모드의 최종 선택 시 사용되며 이전의 두 비용함수에 비해 상대적으로 높은 복잡도를 갖는다. 식 (5)내의 왜곡(Distortion) 값은 각 매크로블록 모드에서의 SNR을 구함으로써 계산할 수 있으며, 율(Rate) 값은 매크로블록의 부호화가 끝나는 시점에서의 압축된 비트열(bitstream)의 비트율을 나타낸다.

율-왜곡 최적화 기법이 사용되는 경우, 참조 소프트웨어는 위의 세 가지 평가함수를 이용하여 다음과 같은 순서로 매크로블록 모드를 결정한다.

Step 1. Mode1, Mode2, Mode3에 대한 참조 프레임 및 움직임 벡터 결정

$$\min_{ref, mv} arg (MVcost(Mode i) + REFcost(Mode i))$$

where,  $i = 1, 2, 3$

Step 2. Mode8 내의 부-매크로블록에 대한 모드 선택

Step 2.1 첫 번째 부-매크로블록에 대한 참조 프레임 및 움직임 벡터 결정

$$\min_{ref, mv} arg (MVcost(Mode i) + REFcost(Mode i))$$

where,  $i = 4, 5, 6, 7$

Step 2.2 첫 번째 부-매크로블록의 모드 선택

$$SubMB Mode = \min(RDCost(Mode4, Mode5, Mode6, Mode7)) \quad (8)$$

Step 2.3 잔여 부-매크로블록에 대해서 Step 2.1~Step 2.2 반복

Step 3. 최종 매크로블록 모드 결정

$$MB Mode = \min(RDCost(SKIP, Mode1 \sim Mode3, Mode8, I16MB, I4MB)) \quad (9)$$

식 (8)과 식 (9)에서  $RDcost(A,B,C)$ 의 표기(表記)는 다음과 같은 의미를 지닌다.

$$RDcost(A,B,C) = (RDcost(A), RDcost(B), RDcost(C)) \quad (10)$$

Step 1에서는 Mode1, Mode2, Mode3 각각에 대해서, 움직임 벡터 비용과 참조 프레임 비용의 합을 최소로 하는 움직임 벡터와 참조 프레임을 구한다. Step 2는 Mode8 내의 부-매크로블록에 대한 모드 선택과정을 나타낸다. Step 2.1에서는 첫 번째 부-매크로블록에 대해서 Mode4부터 Mode7까지 각각의 참조 프레임 및 움직임 벡터를 참조 프레임 비용과 움직임 벡터 비용함수의 합이 최소가 되도록 선택한다. 이후 각 모드에서 선택된 참조 프레임과 움직임 벡터에 대한 윌-왜곡 비용을 구하고 이를 최소화 하는 모드를 선택함으로써 하나의 부-매크로블록 모드가 선택되며 이 과정이 Step 2.2이다. 남아있는 세 개의 부-매크로블록에 대해서도 Step 2.1과 Step 2.2의 과정을 반복함으로써 Mode8 내의 네 개의 부-매크로블록 모드를 선택할 수 있다. Step 2의 과정까지 Mode8 내 부-매크로블록들을 포함하여 Mode1, Mode2, Mode3의 참조 프레임과 움직임 벡터가 결정되면, 이를 이용하여 Step 3에서는 이들의 윌-왜곡 비용과 함께 SKIP 및 공간적 예측을 위한 매크로블록 모드인 I16MB, I4MB의 윌-왜곡 비용을 비교하여 그 값을 최소로 하는 매크로블록 모드를 선택함으로써 최종적으로 하나의 매크로블록 모드가 결정된다.

### III. 제안하는 알고리즘

#### 1. Mode8의 복잡도 감소 알고리즘

참조 소프트웨어에서의 매크로블록 모드 결정과정에서도 볼 수 있듯이 시방향 예측을 위한 5가지의 매크로블록 모드 중 Mode8은 나머지 4가지 모드에 비해 높은 복잡도를 가진다. 이는 움직임 벡터 비용 계산 및 참조 프레임 비용 계산과정의 경우 Mode1~Mode3 및 Mode8에 공통적으로 포함되지만, Mode8의 경우 네 개의 부-매크로블록 모드를 결정하기 위해 Mode1~Mode3에는 없는 부-매크로블록 단위의 윌-왜곡 비용 함수 계산과정을 16회 포함하기 때문이다. 이는 CIF 크기의 영상의 경우, Mode8은 Mode1~Mode3에 비해 프레임 당 6,336회의 추가적인 부-매크로블록 단위의 윌-왜곡 비용함수 계산과정

을 포함함을 의미한다. 따라서 상대적으로 높은 복잡도를 가지는 Mode8의 계산량 감소는 전체 부호기의 복잡도에 큰 영향을 미칠 수 있다.

Mode8의 복잡도 감소를 위해 제안하는 알고리즘은 매크로블록의 나뉘는 방향에 따라 그에 따른 윌-왜곡 비용의 크기가 단조 증가 또는 단조 감소라는 가정에서 출발한다. 그림 5는 가로방향 및 세로방향에 대한 블록 세분화 과정을 나타내며, 식 (11) 및 식 (12)는 윌-왜곡 비용의 단조 증가 및 단조 감소를 수식화 한 것이다.

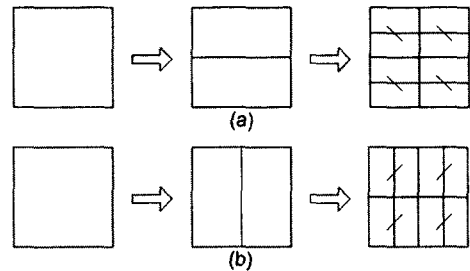


그림 5. 블록 세분화 과정 (a) 가로방향의 블록 세분화 (b) 세로방향의 블록 세분화  
Fig. 5. Inter MB partitions (a) MB partitions in horizontal (b) MB partitions in vertical

$$RDcost(Mode1) \leq RDcost(Mode2) \leq RDcost(Mode8(4|5)) \quad (11)$$

$$RDcost(Mode1) \leq RDcost(Mode3) \leq RDcost(Mode8(4|6)) \quad (12)$$

식 (11)에서 Mode8(4|5)의 표현은 선택가능한 부-매크로블록 모드가 Mode4와 Mode5로 한정됨을 의미하며, 식 (12)의 Mode8(4|6) 역시 선택가능한 부-매크로블록 모드가 Mode4와 Mode6으로 한정됨을 의미한다.

단조 증가 및 단조 감소의 가정은 가령 Mode2의 윌-왜곡 비용이 Mode1의 윌-왜곡 비용보다 작다면 이는 매크로블록이 가로방향으로 나뉘어 짐에 따라 윌-왜곡 비용이 감소하고 있음을 나타내며 따라서 가로방향으로의 추가적인 세분화 과정이 필요함을 의미한다. 반면 Mode2의 윌-왜곡 비용이 Mode1의 윌-왜곡 비용보다 크다면 이는 가로방향으로의 추가적인 블록 세분화 과정에서 Mode1보다 적은 윌-왜곡 비용을 기대할 수 없음을 의미한다. 세로방향의 블록 세분화를 나타내는 식 (12)의 경우에서도 이와 같은 가정을 적용한다.

이러한 가정 하에 Mode8의 복잡도 감소를 위한 제안하는 알고리즘은 다음과 같은 과정으로 수행한다.

Step 1. Mode1~Mode3에 대한  $J_{mode}$  계산 및 움직임 벡터, 참조 프레임 선택

$$\min_{ref, mv} \arg (J_{modei} = MVcost(Modei) + REFCost(Modei)) \quad (13)$$

where,  $i=1,2,3$

Step 2.  $J_{mode}$  값의 대소비교를 통한 Mode8의 계산량 정의

- case 1. If  $J_{mode1} < J_{mode2}$  and  $J_{mode1} < J_{mode3}$  then Mode8(SKIP)
- case 2. If  $J_{mode1} > J_{mode2}$  and  $J_{mode1} < J_{mode3}$  then Mode8(4|5)
- case 3. If  $J_{mode1} < J_{mode2}$  and  $J_{mode1} > J_{mode3}$  then Mode8(4|6)
- case 4. If  $J_{mode1} > J_{mode2}$  and  $J_{mode1} > J_{mode3}$  then Mode8(4|5|6|7)

Step 3. 최종 매크로블록 모드 선택

- case 1.  $\min(RDcost(SKIP, Mode1, I16MB, I4MB)) \quad (14)$
- case 2.  $\min(RDcost(SKIP, Mode2, Mode8(4|5), I16MB, I4MB)) \quad (15)$
- case 3.  $\min(RDcost(SKIP, Mode3, Mode8(4|6), I16MB, I4MB)) \quad (16)$
- case 4.  $\min(RDcost(SKIP, \min(Mode(J_{mode2}, J_{mode3}), Mode8(4|5|6|7), I16MB, I4MB)) \quad (17)$

제안하는 알고리즘은 Mode8의 계산량 정의를 위해 Mode1~Mode3의 비용들을 사용하며, 이때 사용되는 비용은 상대적으로 낮은 복잡도만을 갖는 움직임 벡터 비용 및 참조 프레임 비용을 사용한다.

Step 1에서는 Mode1~Mode3에 대한 움직임 추정 비용과 참조프레임 비용의 합을 나타내는  $J_{mode}$  값을 계산하고, Mode1~Mode3 각각에 대해서  $J_{mode}$ 를 최소로 하는 움직임 벡터 및 참조프레임을 구한다.

Step 2는 앞서 구한 Mode1~Mode3의  $J_{mode}$  값의 비교를 통해 4 가지의 서로 다른 Mode8의 계산량을 정의한다. case 1은 가로방향으로 세분화된 매크로블록의  $J_{mode}$  값과 세로방향으로 세분화된 매크로블록의  $J_{mode}$  값이 모두  $16 \times 16$  크기 매크로블록의  $J_{mode}$  값보다 큰 경우로서, 이 경우는 가로 또는 세로방향으로 더 이상의 세분화 과정이 필요 없는 경우이다. 따라서 Mode8의 움직임 벡터 비용 및 참조 프레임 비용 뿐 아니라 부-매크로블록의 모드를 결정하기 위한 울-왜곡 비용의 계산과정은 모두 생략된다. Mode8(SKIP)은 Mode8의 모든 비용 계산과정이 생략됨을 나타낸다. case 2는 가로방향으로의 매크로블록 세분화를 통해 보다 적은  $J_{mode}$  값이 발생하는 경우로서 이때에는 Mode8 내의 선택가능한 부-매크로블록 모드는 가로방향으로의 나뉜 성분을 가지고 있는 Mode4 및 Mode5로 한정한다. case 3은 세로방향으로의 매크로블록 세분화를 통해 보다 적은  $J_{mode}$  값이 발생하는 경우로서 이때에는 Mode8 내의 선택가능한 부-매크로블록 모드는 세로방향으로의 나뉜 성분을 가지고 있는 Mode4 및 Mode6으로 한정한다. 마지막으로 case 4는 가로 방향으로 세분화된 매크로블록과 세로 방향으로 세분화된 매크로블록의  $J_{mode}$  값이 모두  $16 \times 16$  크기 매크로블록의  $J_{mode}$  값보다 작은 경우로 이때에는 Mode8 내의 부-매크로블록 모드는 모든 세분화 방향을 고려하여 Mode4~Mode7 중에서 선택한다.

Step 3은 참조 소프트웨어의 최종 매크로블록 모드의 선택과 유사하나 이 때 Mode1~Mode3의 울-왜곡 비용 계산은 최소의  $J_{mode}$ 를 갖는 모드로 한정한다. 식 (17)의  $\min(Mode(J_{mode2}, J_{mode3}))$ 는 Mode2와 Mode3 중에서 보다 적은  $J_{mode}$ 를 갖는 모드를 가리킨다. 표 1은 QCIF 크기의 테스트 시퀀스 100 프레임 을 IPPP 구조로 부호화 하는 경우, 부-매크로블록의 모드 결정을 위해 사용되는 울-왜곡 비용의 계산량 감소정도를 정량적으로 표현한다. QCIF 크기 영상의 경우 Mode8 내의 부-매크로블록 모드 결정에 사용되는 울-왜곡 비용 계산 회수는 프레임 당 1,584회이다. 따라서 IPPP 구조로 부호화 한 경우 99개의 P 프레임에 대해서 Mode8의 부-매크로블록 모드 선택에 사용되는 울-왜곡 비용의 전체 계산회수는 156,816회이다.

표 1. 참조 프레임 수에 따른 Mode8내의 부-매크로블록의 선택을 위해 사용되는 RD cost 계산량의 감소율  
 Table 1. The reduction rate of RD cost calculations to select subMB modes depending on the number of reference frames in Mode8

Test Sequences (QCIF)	참조 프레임수	총 계산회수	계산량 감소율 (%)
Akiyo	1	26,576	83.05
	5	21,256	86.45
Claire	1	27,376	82.54
	5	23,680	84.90
Container	1	17,896	88.59
	5	17,376	88.92
News	1	30,256	80.71
	5	28,432	81.87
Coastguard	1	59,760	61.89
	5	57,016	63.64
Stefan	1	56,472	63.99
	5	66,016	57.90
Table	1	63,064	59.78
	5	61,872	60.54
Trevor	1	54,800	65.05
	5	52,968	66.22

표 1에서 보는 바와 같이 제안하는 Mode8 복잡도 감소 알고리즘은 영상에 따라서 적게는 58%에서부터 최대 89%까지의 울-왜곡 비용 계산량 감소 효과가 있는 것으로 나타난다.

2. I4MB 모드의 복잡도 감소 알고리즘

앞서 기술한 바와 같이 P 프레임 내 하나의 매크로블록 모드는 시방향 예측을 위한 5개의 모드와 공간 내 예측을 위한 2개의 모드 중에서 선택 될 수 있다. 공간 영역에서의 예측을 사용하는 I16MB 모드와 I4MB 모드 중에서 I16MB의 모드는 16×16 크기의 매크로블록에 대해서 단지 4개의 예측 모드 (Vertical-Plane)를 가지는데 반해 I4MB 모드는 4×4 크기의 블록에 대해서 9개의 예측모드를 가지고 있기 때문에 I16MB 모드에 비해 상대적으로 높은 복잡도를 갖는다. 제안하는 알고리즘은 I16MB의 울-왜곡 비용과 I4MB의 울-왜곡 비용의 상관성을 이용한다. I4MB 모드는 I16MB 모드에 비해 더욱 세분화된 블록의 크기에 대하여 더욱 많은 방향성을 고려함으로써 항상 I16MB 모드보다 작은 예측오차만을 가진다.

이는 I4MB 모드의 세분화된 블록크기와 방향성이 I16MB 모드에서 고려하고 있는 모든 예측 방향들을 포함하고 있기 때문이다. 하지만 I16MB 모드가 단 하나의 예측 모드(2 비트)만을 부가정보로서 가지는 반면 I4MB 모드의 경우에는 16개의 예측모드(16×3 비트)를 부가정보로 발생한다. I4MB 및 I16MB 모드의 예측오차 차이는 매크로블록 내에 I16MB 모드에서 고려하지 못한 방향성의 에지 성분이 있는 경우 또는 세세한 에지의 방향이 매크로블록 내에 존재할 때 발생한다. 하지만 예측 모드의 부가정보량을 고려해 볼 때 전체 울-왜곡 비용의 차이는 그리 크지 않으며 이와 같은 I16MB와 I4MB의 울-왜곡 비용의 유사성을 근거로 공간 내 예측을 위한 I4MB 모드의 계산량을 줄일 수 있다.

제안하는 알고리즘은 매크로블록 모드의 결정시 앞서 구한 시방향 예측을 위한 매크로블록 모드들의 울-왜곡 비용과 상대적으로 낮은 복잡도만을 가지는 I16MB 모드의 울-왜곡 비용을 비교함으로써 I4MB 모드의 울-왜곡 비용함수의 계산 여부를 결정한다. I4MB 모드의 울-왜곡 비용함수 계산과정이 생략되는 경우에는 I4MB에서 9가지 예측방향으로의 공간 내 예측과정 또한 생략된다. I4MB 모드의 계산 여부를 결정하는 수식은 아래와 같다.

$$\min(RDcost(SKIP, Mode1, Mode2, Mode3, Mode8)) \cdot k < RDcost(I16MB) \quad (18)$$

식 (18)에서 k는 비례상수이다. 시방향 예측을 위한 5가지의 매크로블록 모드의 울-왜곡 비용을 구하고, 그중 최소의 울-왜곡 비용 값에 비례상수 k를 곱하여 이 값이 I16MB의 울-왜곡 비용 값보다 작은 경우에는 I4MB 모드에서 최소의 울-왜곡 비용 값이 나타날 것이라고는 기대하기 어렵다. 이는 I16MB 및 I4MB가 공간 내 방향성을 고려하여 예측을 수행하며, 단지 예측 단위와 고려하는 방향성의 차이만을 가지기 때문에 울-왜곡 측면에서의 비용은 상당한 상관성을 가지고 있다는 가정에 근거한다. 따라서 식 (18)을 만족하는 경우 I4MB의 울-왜곡 비용함수 계산 과정 및 9가지 방향의 공간 내 예측과정은 모두 생략된다.

표 2는 I16MB의 울-왜곡 비용과 I4MB의 울-왜곡 비용 값 사이에 큰 상관성이 존재한다는 가정이 정당함을 나타내며 각 비례 상수 k에 따른 오류 확률을 나타낸다. 오류확률은 식 (18)을 사용하여 계산과정이 생략된 I4MB 모드에서 최소의 울-왜곡 비용 값이 발생할 확률을 나타낸다.



표 2. k 값에 따른 오류 확률  
Table 2. The missing rates depending on k

Test Sequences (QCIF)	오류확률 (%) k = 1.3	오류확률 (%) k = 1.5	오류확률 (%) k = 1.7
Akiyo	7.72	1.46	0.00
Claire	12.03	3.44	0.08
Container	2.69	0.98	0.85
News	5.76	0.57	0.51
Coastguard	1.99	0.44	0.17
Stefan	2.79	0.17	0.00
Table	1.08	0.04	0.00
Trevor	9.03	0.89	0.06

표 2의 결과는 비례상수 k가 1.5의 작은 값을 갖는 경우에도 오류확률은 단지 3.5% 이내이며 평균 오류 확률은 약 1%에 그친다. 이는 영상 내 평균 99%의 매크로블록에서 I16MB 및 I4MB의 울-왜곡 비용 값이 1.5배 이내의 차이를 보임을 나타낸다.

하나의 매크로블록에 대한 I4MB의 울-왜곡 비용 함수의 계산 회수는 매크로블록의 위치에 따라 다르게 나타난다. 이는 이웃한 블록의 존재 유무에 따라 색차성분의 공간적 예측 방향(DC-Plane)이 제한되기 때문이다. 상측 또는 좌측의 이웃한 매크로블록이 모두 없는 경우에는 I4MB의 울-왜곡 비용 함수는 단 한번만 계산을 하며, 상측 및 좌측 중 하나의 이웃한 매크로블록만 있는 경우에는 두 번의 울-왜곡 비용 함수의 계산회수를 갖는다. 상측 및 좌측의 이웃한 매크로블록이 모두 존재하는 경우에는 모든 색차성분의 예측방향에 따라 수행되며, 따라서 그런 매크로블록은 네 번의 울-왜곡 비용함수의 계산을 수행한다. 예를 들어, 울-왜곡 최적화 기법이 사용되는 경우 QCIF 크기의 영상은 프레임 당 357회의 I4MB 울-왜곡 비용함수를 계산한다.

표 3은 100 프레임의 QCIF 크기의 테스트 시퀀스를 울-왜곡 최적화 기법을 사용하여 IPPP 구조로 부호화 한 경우, I4MB 모드의 울-왜곡 비용함수 계산 회수와 계산량의 평균 감소율을 나타낸다. 이때 비례상수 k는 1.5를 사용하였다. 참조 소프트웨어의 경우 99개의 P 프레임에 대해서 35,343회의 I4MB 울-왜곡 비용함수를 계산한다.

표 3에서 보는 바와 같이 제안하는 I4MB의 복잡도 감소 알고리즘을 통해서 I4MB의 울-왜곡 비용함수의 계산회수는 적게는 69.3%부터 최대 99.7%까지 감소하는 것으로 나타났으며 위의 테스트 시퀀스들에

표 3. I4MB의 울-왜곡 비용함수의 계산량 감소율 (k=1.5)  
Table 3. The reduction rate of RD cost calculations of I4MB for k=1.5

Test Sequences (QCIF)	참조 프레임 수		평균 감소율 (%)
	1 장	5 장	
Akiyo	88	91	99.7
Claire	2,822	3,303	91.3
Container	6,566	7,066	80.7
News	2,225	2,231	93.7
Coastguard	10,838	10,828	69.3
Stefan	3,038	2,713	91.9
Table	8,263	7,860	77.2
Trevor	3,463	3,489	90.2

대한 평균 감소율은 약 86.7%인 것으로 나타난다.

#### IV. 실험 및 결과

매크로블록 모드의 고속 선택 기법을 통한 부호기 전체의 복잡도를 감소시키기 위해 논문에서 제안한 두 가지 알고리즘은 각각 독립적으로 적용가능하다. 실험에서는 두 가지의 알고리즘을 모두 적용하였으며 실험 환경은 다음과 같다.

- CPU : 인텔 펜티엄 4 2.0GHz
- 메모리 : 512 MHz
- 운영체제 : Windows XP Pro.
- 적용 코덱 : JM 8.4
- 부호화 구조 : IPPP
- 양자화 파라미터 : 28
- 울-왜곡 최적화 : 사용
- 움직임 추정기법 : 전역 검색 (Full Search)

실험은 100 프레임의 QCIF 크기의 테스트 시퀀스를 사용하였으며, 테스트 시퀀스의 선택은 압축된 비트열의 비트율에 따라 8개를 선정하여 사용하였다. 압축 비트열의 비트율이 100 Kbps 이하의 시퀀스로는 Akiyo, Claire, Container, News를 선택하였으며, 100 Kbps 이상의 시퀀스로는 Coastguard, Stefan, Table, Trevor를 선택하였다.

그림 6은 한 장의 참조프레임을 사용하는 경우, 참조 소프트웨어와 제안한 방법의 전체 부호화 시간을 나타낸다.

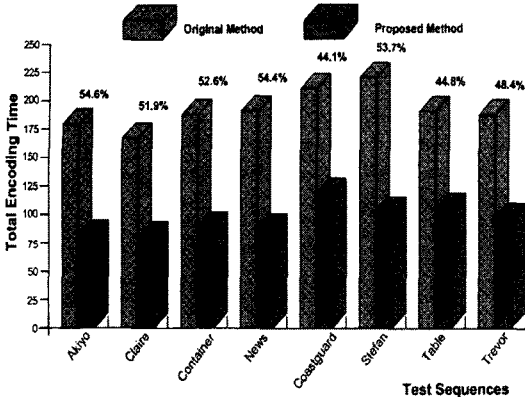


그림 6. 전체 부호화 시간 (참조 프레임이 한 장인 경우)  
Fig. 6. Comparison of encoding time of sequences. (one reference frame)

결과에서 보듯이 압축된 비트열의 비트율 및 영상 내 움직임 정도에 관계없이 적게는 44.1%에서부터 최대 54.6%까지 부호화 시간이 단축되었음을 볼 수 있다. 반면 비트율 및 PSNR[dB]은 미미(微微)한 변화만을 가지는데 이를 표 4와 표 5에 나타내었다.

표 4와 표 5의 결과에서 보는 바와 같이 한 장의 참조 프레임을 사용하는 경우 비트율은 평균 1.3% 증가만을 보이며 PSNR은 평균 0.015dB 감소하는 것으로 나타났다. 표 5에서 Akiyo 및 Claire 영상의 경우 제안된 방법에서 오히려 높은 PSNR을 보이는 것은 다른 영상들에 비해서 비교적 높은 비율로 비트율이 증가했기 때문이다.

표 4. 압축 비트열의 비트율(Kbps)의 비교 (참조 프레임이 한 장인 경우)

Table 4. The bitrates resulting from the proposed scheme versus reference software (one reference frame)

Test Sequences	Original Method	Proposed Method	Comparison (%)
Akiyo	30.40	30.95	Δ 1.8
Claire	32.06	32.89	Δ 2.6
Container	40.16	40.74	Δ 1.4
News	75.84	76.75	Δ 1.2
Coastguard	249.00	251.28	Δ 0.9
Stefan	379.26	380.86	Δ 0.4
Table	193.75	196.49	Δ 1.4
Trevor	132.49	133.58	Δ 0.8

표 5. 압축 비트열의 비트율(Kbps)의 비교 (참조 프레임이 한 장인 경우)

Table 5. PSNRs resulting from the proposed scheme versus reference software (one reference frame)

Test Sequences	Original Method	Proposed Method	Comparison [dB]
Akiyo	38.18	38.22	Δ+0.04
Claire	39.65	39.7	Δ+0.05
Container	36.07	36.06	Δ-0.01
News	36.65	36.64	Δ-0.01
Coastguard	33.93	33.89	Δ-0.04
Stefan	34.22	34.15	Δ-0.07
Table	35.47	35.46	Δ-0.01
Trevor	36.40	36.33	Δ-0.07

그림 7은 다섯 장의 참조프레임을 사용하는 경우, 참조 소프트웨어와 제안한 방법의 전체 부호화 시간을 나타낸다.

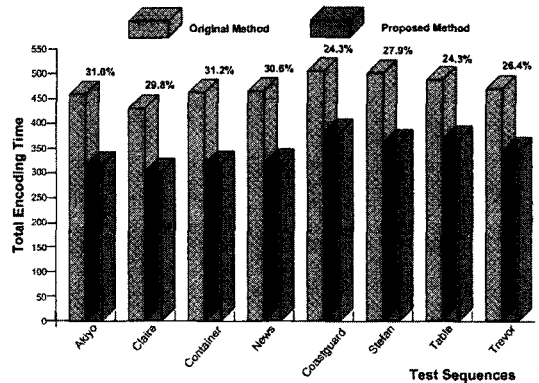


그림 7. 전체 부호화 시간 (참조 프레임이 다섯장인 경우)  
Fig. 7. Comparison of encoding time of sequences. (five reference frames)

다섯 장의 참조프레임을 사용하는 경우 약 30%의 부호화 시간 단축을 나타낸다. 한 장의 참조프레임을 사용하는 경우보다 다소 낮은 부호화 시간의 단축을 보이는 것은 전체 부호화 과정에서 움직임 추정과정이 차지하는 비율이 참조프레임의 수에 따라 다르게 나타나기 때문이다. 전체 부호화 시간 중 대부분은 매크로블록 모드 선택과정과 움직임 추정과정이 차지하는데 이중 움직임 추정과정이 차지하는 비율을 표 6에 나타내었다. 표 6에서 보는 바와 같이 한 장의 참조프레임을 사용하는 경우 전체 부호화 시간 중 움직임 추정과정이 차지하는 비율은 약 27%인 반면 다섯 장의 참조프레임을 사용하는 경우에는 51%이다. 이는 상대적으로 참조프레임수가 늘어날수록 전체 부호화 시간 중 매크로블록 모드 선택과정이 차지하는

비율이 감소함을 의미한다. 따라서 고속 매크로블록 모드 선택 기법을 사용한 경우, 적은 참조프레임을 사용할 때 보다 높은 성능을 보이는 것은 이 때문이다.

표 6. 참조 프레임 수에 따른 전체 부호화 시간 중에 움직임 추정 시간이 차지하는 비율  
Table 6. Of the total encoding time, the rates of ME time depending on the number of reference frames

Test Sequences (QCIF)	The number of reference frame	Total (sec)	ME (sec)	ME/Total (%)
Akiyo	1	179.7	51.4	28.6
	5	459.1	238.6	52.0
Claire	1	167.3	48.9	29.2
	5	430.6	228.3	53.0
Container	1	187.9	50.1	26.7
	5	462.8	236.3	51.0
News	1	191.1	50.2	26.3
	5	465.6	236.8	50.9
Coastguard	1	212.1	54.4	25.7
	5	505.9	248.2	49.1
Stefan	1	220.8	51.3	23.2
	5	501.5	241.7	48.2
Table	1	190.7	51.6	27.0
	5	488.0	244.4	50.1
Trevor	1	187.7	50.5	26.9
	5	469.9	238.5	50.8

표 7과 표 8은 다섯 장의 참조프레임을 사용한 경우 압축된 비트열의 비트율 및 PSNR을 나타낸다.

표 7. 압축 비트열의 비트율[Kbps]의 비교 (참조 프레임이 다섯 장인 경우)  
Table 7. The bitrates resulting from the proposed scheme versus reference software (five reference frames)

Test Sequences	Original Method	Proposed Method	Comparison (%)
Akiyo	29.47	30.36	Δ3.0
Claire	31.74	32.51	Δ2.4
Container	36.23	36.81	Δ1.6
News	75.36	76.29	Δ1.2
Coastguard	244.85	246.30	Δ0.6
Stefan	356.07	358.83	Δ0.8
Table	190.22	192.25	Δ1.1
Trevor	133.77	134.82	Δ0.8

표 8. PSNR[dB]의 비교 (참조 프레임이 다섯 장인 경우)  
Table 8. PSNRs resulting from the proposed scheme versus reference software (five reference frames)

Test Sequences	Original Method	Proposed Method	Comparison [dB]
Akiyo	38.25	38.30	Δ+0.05
Claire	39.71	39.73	Δ+0.02
Container	36.12	36.12	Δ0.00
News	36.70	36.67	Δ-0.03
Coastguard	34.01	33.98	Δ-0.03
Stefan	34.35	34.30	Δ-0.05
Table	35.55	35.54	Δ-0.01
Trevor	36.43	36.40	Δ-0.03

결과에서 보는 바와 같이 다섯 장의 참조 프레임을 사용하는 경우에도 한 장의 참조 프레임을 사용하는 경우와 유사하게 비트율은 평균 1.4% 증가만을 보이며 PSNR은 평균 0.01dB 감소하는 것으로 나타난다. 이 경우도 역시 Akiyo 및 Claire 영상의 경우 제안된 방법에서 오히려 높은 PSNR을 보이는 것은 한 장의 참조프레임을 사용한 경우와 마찬가지로 다른 영상들에 비해서 비교적 높은 비율로 비트율이 증가했기 때문이다.

## V. 결론

H.264/AVC 비디오 부호화 표준은 새로운 부호화 도구들의 사용으로써 보다 높은 부호화 효율을 가지지만 부호기의 복잡도는 상당히 증가한다. 다양한 블록 크기로 움직임 추정을 수행하는 VBS의 선택과정과 1/4 화소단위의 움직임 추정 방식은 이러한 부호기 복잡도 증가의 주요 원인이다.

제안한 알고리즘은 VBS 선택과정의 고속 처리 기법으로써 시방향의 움직임 추정에 사용되는 다양한 블록 크기들 중 상대적으로 높은 복잡도를 가지는 Mode8의 비용값 계산과정과 공간내 예측에 사용되는 매크로블록 모드들 중 상대적으로 높은 복잡도를 가지는 I4MB의 비용값 계산과정을 효율적으로 줄임으로써 전체 부호화 시간을 단축시킨다.

실험결과 Mode8의 비용값 산출과정에 사용되는 윌-왜곡 비용함수의 계산회수는 8개의 실험 영상에 대해서 평균 73% 감소하였으며, I4MB의 경우 평균 86% 감소하는 것으로 나타났다. 또한 전체 부호화 시간 또한 참조프레임이 한 장인 경우 약 50% 단축되었으며 참조프레임이 다섯 장인 경우 약 30%의 시간이 단축되었다. 반면 비트율은 평균 1.5% 증가하는

데 그쳤으며, PSNR 또한 평균 0.012dB의 미미한 감소만을 가진다.

제안하는 알고리즘은 임계값을 사용하지 않아 입력 영상에 따라 의존적이지 않으며, 낮은 비트율과 높은 비트율, 모두에서 비슷한 성능을 보인다. 제안하는 알고리즘은 움직임 추정 기법을 전역 검색으로 제한하였으나 최근의 여러 고속 움직임 추정 기법들과 함께 사용함으로써 H.264 부호기를 통한 실시간 부호화를 기대할 수도 있다.

### 참 고 문 헌

[1] JVT G050r1, "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC)," May 2003.

[2] Thomas Wiegand, Gary J. Sullivan, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, pp. 560-576, July 2003.

[3] Thomas Wiegand, Heiko Schwarz, Anthony Joch, Faouzi Kossentini, "Rate-Constrained Coder Control and Comparison of Video Coding Standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, pp. 688-703, July 2003.

[4] Peng Yin, Hye-Yeon Cheong, Tourapis, Alexis Michael Tourapis, Jill Boyce, "Fast Mode Decision and Motion Estimation for JVT/H.264," *Image Processing, 2003 International Conference on*, vol. 3 pp. 853-856, Sept. 2003.

[5] Yu-Wen Huang, Bing-Yu Hsieh, Tu-Chih Whang, Shao-Yi Chien, Shyh-Yih Ma, Chun-Fu Shen, Liang-Gee Chen, "Analysis and Reduction of Reference Frames for Motion Estimation in MPEG-4 AVC/JVT/H.264," *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, vol. 3 pp. 145-148, April 2003.

[6] A. Ahmad, N. Khan, S. Masud, M.A. Maud, "Efficient block size selection in H.264 video coding standard," *Electronics Letters*, vol. 40 pp. 19-21, Jan. 2004.

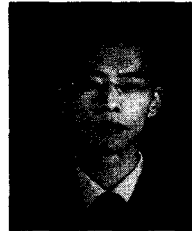
[7] JVT Model downloaded form

<http://bs.hhi.de/~suehring/tml/download/jm84.zip>, accessed 2004

[8] Iain E. G. Richardson, *H.264 and MPEG-4 Video Compression*, Wiley, Inc., 2003.

김 동 형(Donghyung Kim)

정회원



1999년 2월 : 충북대학교 전자공학과 졸업  
2001년 8월 : 충북대학교 전자공학과 석사  
2002년 3월~현재 : 한양대학교 전자통신전파공학과 박사과정

<관심분야> 영상처리 및 영상압축

정 제 창(Jechang Jeong)

정회원



1980년 2월 : 서울대학교 전자공학과 졸업  
1982년 2월 : KAIST 전기전자공학과 석사  
1990년 : 미국 미시간대학 전기공학과 공학박사

1980~1986 : KBS 기술연구소 연구원 (디지털 TV 및 뉴미디어 연구)

1990~1991 : 미국 미시간대학 전기공학과 연구교수 (영상 및 신호처리 연구)

1991~1995 : 삼성전자 멀티미디어 연구소(MPEG, HDTV, 멀티미디어 연구)

1995~현재 : 한양대학교 전자전기컴퓨터공학부 교수 (영상통신 및 신호처리 연구실)

1998년 11월 27일 : 과학기술자상 수상

1998년 12월 31일 : 정보통신부장관상 표창

<관심분야> 영상처리 및 영상압축