Journal of Korea Multimedia Society Vol. 11, No. 10, October 2008(pp. 1359-1365)

H.264/AVC에서 효율적인 움직임 벡터와 모드 정보의 압축

이동식†, 김영모†

요 약

H.264에서 채택하고 있는 기술들로 인해 H.264의 헤더는 이전 표준안들에 비해 전체 비트 열에서 더 많은 비율을 차지하기 때문에, H.264의 헤더를 압축하기 위한 새로운 기술이 필요하다. H.264에서는 구문 요소를 일원화하여 부호화하는데, 부호화할 요소들의 발생 분포를 고려하지 않고 기존의 Exp-Golomb 방식을 이용하기 때문에 가변 길이 부호화 관점에서 매우 비효율적이다. 헤더의 대부분을 매크로 블록 타입과 움직임 벡터 차이 값이 차지하고 있으며, 본 논문에서 분석한 H.264의 헤더에서의 중복은 다음과 같은 세 가지이다. 매크로 블록 타입에서 자주 발생하는 부호와 그렇지 않는 부호가 있으며, 매크로 블록 모드가 8일 때, 네 개의 서브 매크로 블록 타입들이 모두 전송된다. 그리고 마지막으로 움직임 벡터 차이 값에서 같은 값(특히 '0')들이 발생한다. 본 논문에서는 타입 코드와 쿼드트리를 사용하는 알고리즘을 제안하고 있으며 헤더에서의 반복되는 정보를 이 두 가지 구조들을 가지고 표현한다. 타입 코드는 발생하는 매크로 블록의 모양을 나타내며, 쿼드트리는 움직임 추정 나무 구조를 나타낸다. 실험의 결과에서 제안하는 알고리즘이 JM12.4에 비해 최대 32.51% 비트율 감소를 보여준다.

Efficient Coding of Motion Vector and Mode Information for H.264/AVC

Dong-Shik Lee†, Young-Mo Kim††

ABSTRACT

The portion of header in H.264 gets higher than those of previous standards instead of its better compression efficiency. Therefore, this paper proposes a new technique to compress the header of H.264. Unifying a sentence elementary in H.264, H.264 does not consider the distribution of element which be encoded and uses existing Exp-Golomb method, but it is uneffective for variable length coding. Most of the header are block type(s) and motion vector difference(s), and there are redundancies in the header of H.264. The redundancies in the header of H.264 which are analyzed in this paper are three. There are frequently appearing symbols and non-frequently appearing symbols in block types. And when mode 8 is selected in macroblock, all of four sub-macroblock types are transferred. At last, same values come in motion vector difference, especially '0.' This paper proposes the algorithm using type code and quadtree, and with them presents the redundant information of header in H.264. The type code indicates shape of the macroblock and the quadtree does the tree structured motion compensation. Experimental results show that proposed algorithm achieves lower total number of encoded bits over JM12.4 up to 32.51% bit reduction.

Key words: H.264(H.264), Quadtree(쿼드트리), motion vector(움직임 벡터), type code(타입 코드), header information(헤더 정보)

※ 교신저자(Corresponding Author) : 이동식, 주소 : 대구광역시 북구 산격동 1370(702-701), 전화 : 053)940-8641
FAX : 053)950-5541, E-mail : lds@ee.knu.ac.kr
접수일 : 2008년 1월 24일, 완료일 : 2008년 8월 25일

† 준회원, 경북대학교 전자전기컴퓨터학부
†† 경북대학교 전자전기컴퓨터학부 교수
(E-mail : ymkim@ee.knu.ac.kr)

1. 서 론

최근, 동영상 부호화를 위해 ISO/IEC의 Moving Picture Experts Group (MPEG)과 ITU-T의 Video Coding Experts Group (VCEG)는 공동으로 새로운 표준안을 발표하였는데, 이 표준안을 MPEG-4 Part 10 Advanced Video Coding (AVC) 혹은 H.264 recommendation이라고 부른다[1]. 멀티미디어 압축과 전송 분야에서 널리 사용되고 있는 MPEG-2가 발표된 이후 움직임 추정/보상과 압축을 위한 블록 크기는 작아지는 경향이 있는데, MPEG-2는 16×16 블록을 사용하고 MPEG-4는 16×16과 8×8 블록을 사용할 수 있다. 이들 표준들에서는 움직임 추정/보상으로 발생하는 움직임 벡터 차이(motion vector difference : mvd)를 포함하는 헤더의 크기가 전체 비트열에 비해 무시할 만 수준이었기 때문에, 압축 시 헤더를 고려하지 않는다. 일반적으로 mvd 정보는 헤더의 전체 크기에서 가장 큰 부분을 차지하고 있으며, MPEG-2에서 헤더의 크기는 전체 비트열의 10% 정도이다.

하지만, H.264는 MPEG-2/4 보다 더 작은 크기인 4×4 블록을 사용하는 mv와 다중 참조 영상을 사용하기 때문에 더 정밀한 움직임 보상을 수행한다. H.264에서 선택된 모드가 4×4 블록이라면, 최대 16개의 움직임 벡터와 4개의 참조 영상들이 전송된다. 더욱이, H.264에서 채택하고 있는 모드 선택 기법으로 인해 매크로 블록 타입까지 결과 비트 열에 전송되고 있으며, 이런 기술들로 인해 헤더가 전체 비트열의 50%까지 차지한다. 하지만, H.264의 기술들은 단지 잔여 데이터의 압축에 주안점을 두고 있으며, 이전 표준안에 비해 중요해진 헤더는 크게 고려되고 있지 않다. H.264에서는 Exp-Golomb 부호화 방식을 사용하는 가변 길이 부호화를 사용하여 구문 요소를 부호화하고 있다. 그러나 부호화 과정에서 부호화할 부호의 발생 분포를 고려하지 않고 기존의 Exp-Golomb 방식을 이용하기 때문에 가변 길이 부호화 관점에서 매우 비효율적이다[2,3].

본 논문에서는 이전 논문들에서 고려하고 있지 않는 헤더에서 발생할 수 있는 다양한 중복을 제거하여, 압축 효율을 높이는 알고리즘을 제안한다. 헤더에서는 (매크로) 블록의 블록 타입과 mvd가 저장되는데, 여기에서 발생할 수 있는 중복성에 대한 분석을 하고 이 중복을 제거하는 효율적인 압축 구조를 제안한다. 본 논문은 H.264에서 이러한 문제점을 해결하고 효율적인 헤더 압축을 위해, 타입 코드와 쿼드트리를 사용하여 블록 타입 (매크로 블록의 mb_type와 서브 매크로 블록의 sub_type)과 mvd를 표현한다.

H.264의 헤더를 압축하기 위한 여러 연구가 있었지만 이들 연구들은 모드 선택에서 mv의 개수나 비트 율을 줄이는 방향으로 접근을 하고 있다. Kwon 등[4,5]은 그들의 연구에서 헤더 정보를 나타내는 모델을 제안하였는데, 이 모델은 '0'이 아닌 움직임 벡터 요소와 움직임 벡터의 개수에 대한 함수로 표현된다. Wong 등[6]은 H.264에서 헤더를 압축하기 위해서 헤더의 크기를 고려하는 엔트로피 부호화 방식을 제안하였으며, 결과에서 10%의 비트 열 압축이나 같은 비트 율에서 0.3-0.4 dB 향상을 보여주었다. 하지만 위의 논문들에서는 헤더 정보에서 발생하는 중복에 대한 분석을 보여주지 못하고 있으며 이것에 대한 해결방안을 제시하지 못하고 있다.

본 논문의 나머지 부분은 다음과 같이 구성된다. H.264의 헤더에 존재하는 블록 타입과 mvd의 중복성이 2장에서 설명된다. 3장에서는 제안하는 알고리즘과 사용되는 쿼드트리와 타입 코드를 설명한다. 실험 결과가 4장에서 주어지며 제안하는 알고리즘의 성능을 보여준다. 결론은 5장에서 주어진다.

2. H.264 헤더에서의 mb_type 과 mvd

H.264의 매크로 블록 헤더는 mb_type, inter_mb (reference frame과 mvd), cbp와 delta_quant_mb를 포함하고 있고, mb_type과 mvd가 헤더의 대부분을 차지하기 때문에 본 논문에서는 이 두 부분들을 전체 헤더로 인식한다. 그림 1은 Foreman 영상 열중의 다섯 Prediction (P) Type 영상들에서 mb_mode, inter_mode와 전체 비트열의 비트 율들을 보여주고 있으며, 그림 1에서 평균적으로 헤더가 전체 비트열의 45%를 차지하는 것을 보여주고 있다. 일반적으로 헤더의 크기와 비율은 영상의 내용과 양자화 파라미터에 의존하며, 영상 내용이 복잡해질수록 헤더의 크기가 커지지만, 영상의 내용이 간단할 때는 오히려 헤더의 비율이 커진다.

H.264는 압축 과정에서 이전 표준안에서 사용하

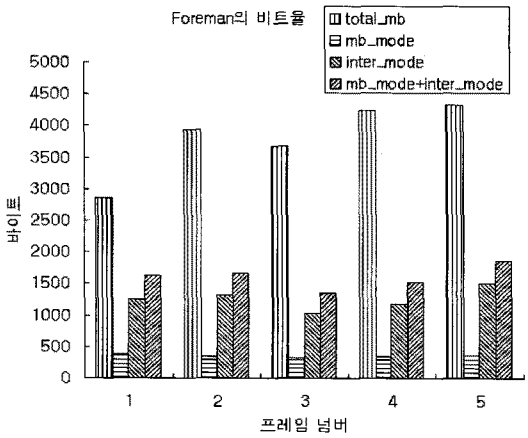


그림 1. Foreman 영상 열의 다섯 P type 영상들에서 mb_mode, inter_mode와 전체 비트열의 비트 율 (QP : 28)

지 않는 모드 선택이라는 과정을 추가하였으며, 이 모드 선택 과정에서 비트율과 왜곡을 동시에 고려하고 있기 때문에 이것을 비트율-왜곡 최적화(Rate-distortion optimization : RDO)라고 한다. RDO의 비트율-왜곡 비용(cost function)은 아래 수식 1과 같다.

$$J = D + \lambda R(s, c, MODE \mid QP) \quad (1)$$

여기서 D는 블록의 예측 왜곡, λ는 라그랑지안 멀티플라이어를 나타내며, R은 비트 율이다. s와 c는 각각 원 입력 신호와 재생 신호를 나타낸다. $MODE$는 매크로 블록의 모드이며, QP는 양자화 파라미터다. H.264는 비디오 압축의 최적화를 위해서 RDO을 사용하는 화면 간 모드를 선택한다. 화면 간 모드에서 사용가능한 모든 모드는 수식 2와 같다.

$$Mode_{\cap} = \text{SKIP}, \cap, \text{I4MB}, \text{I16MB} \quad (2)$$

H.264의 화면 내 모드에는 두 가지 옵션들이 있다: I4MB와 I16MB. I4MB 예측 모드에는 4×4 블록을 사용하는 여덟 방향 모드와 DC 예측 모드가 있다. I4MB와 다르게, 전체 16×16 매크로 블록을 사용하여 예측하는 I16MB 모드에는 세 방향 모드와 DC 예측 모드가 존재한다. H.264에는 화면 간 부호화 효율을 높이기 위해, 가변 블록 크기 움직임 예측을 사용하는 일곱 가지의 화면 간 예측들이 있다. 16×16 매크로 블록은 다음과 같이 분할될 수 있다. 16×16, 16×8, 8×16과 8×8 서브-매크로 블록. 만약 8×8 서브-매크로 블록 분할 방식이 선택되어진다면, 4개의 8×8 서브-매크로 블록들은 다음과 같은 방식으로 각각 분할될 수 있다 : 8×8, 8×4, 4×8, 4×4 블록. 화면 내 모드에서는 위와 같이 7가지 옵션에 따라 RDO를 수행하고, 그 중에서 최적화된 옵션을 선택한다. 마지막으로, 참조 영상이 이전 영상이고 16×16 매크로 블록이 사용되고 mv값이 예측 값과 같다면 SKIP 모드가 선택된다. 최종적으로 SKIP모드, 화면 내 모드와 화면 간 모드에서 가장 최적화된 모드를 선택한다.

위에서 언급한 모든 모드들에 대해 부호화를 수행한 후, 가장 작은 비트율-왜곡 비용을 가지는 모드가 매크로 블록을 부호화하는 최종 모드로 결정된다. 비트 열에서 매크로 블록의 헤더는 그림 2처럼 이루어져 있다.

매크로 블록 헤더는 매크로 블록과 서브-매크로 블록의 내용이 반복적으로 이루어져 있다. 이러한 구조를 나무 구조 움직임 보상이라고 하며, 이러한 구조는 기본적으로 반복이 발생할 수 있다. 예를 들어, x축에서 mvd가 0001 이면, 8 4444 0001 가 비트 열에 주어지는데 (그림 3), 이 예는 비트 열 코드에서 많은 중복이 발생할 수 있다는 것을 보여주는 동시에 제거 가능하다는 것도 보여 준다. 4는 sub_type에서 세 번 반복되고 0은 mvd에서 두 번 반복되어 전체 비트 열에서 발생하는 중복성이 크다는 것을 확인할 수 있다.

더욱이, 정사각형 모양을 가지는 1, 8, 4 모드는 자연 영상 열에서 자주 발생하는 모드들이다 (표 1). 본 논문에서는 헤더에서 발생하는 중복성에 주안점을 두고 이 문제를 해결하기 위해 헤더의 크기를 줄

mb_type	sub_mb_type	ref_idx0	ref_idx1	ref_idx2	ref_idx3	mv0	mv1	mv2	mv3

그림 2. 매크로 블록 헤더

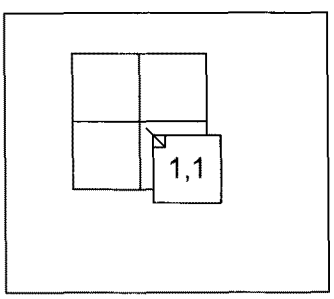


그림 3. 네 번째 블록에서의 움직임 추정 : mvd_x = 1, mvd_y = 1

표 1. 176×144 해상도의 Foreman 두 번째 영상에서 선택된 모드들 (P type, QP : 28)

mode	PSKIP	P16×16	P16×8	P8×16	P8×8	I4MB	I16MB	I8MB
	0	1	2	3	8	9	10	13
발생 횟수	22	20	7	13	23	5	0	9

이는 알고리즘을 제안한다.

3. 쿼드트리와 타입 코드를 이용한 헤더 압축

화면 간 모드의 매크로 블록 타입에서 사용가능한 블록 크기는 16×16부터 4×4까지이며 (그림 4), 전체 일곱 개의 모드를 가진다. 일반적으로 큰 블록은 배경화면 같은 동질 영역에서 선택되어지고, 작은 블록은 서로 다른 방향성을 가지는 비동질 영역에서 선택되어진다.

먼저, 우리는 H.264 헤더에서의 세 가지 중복을 자세하게 분석한다. 1) 매크로 블록과 서브 매크로 블록에서 발생할 수 있는 모양은 세 가지가 있다. 이들은 정사각형(16×16, 8×8와 4×4)과 두 가지 모양의 직사각형(16×8, 8×4와 8×16, 4×8)이며, 자연영상에서 정사각형은 다른 두 직사각형에 비해 높은 발생확률을 가진다(표 1). 그러므로 자주 발생하는 타입들에는 적은 비트 율을 제공하여 압축 성능을 높일 수 있다 (예를 들어, 엔트로피 부호화). 2) 매크로 블록이 서브 매크로 블록으로 분할될 때, 네 개의 모든 서브 매크로 블록 타입들을 비트 열에 전송한다. 이 때, 네 개의 서브 매크로 블록이 서로 다른 타입인 경우가 적으며, 같은 타입을 가지는 블록들이 존재할 가능성이 크다. 3) mvd가 같은 값을 가질 경우의 수가 크다. 특히 '0'의 값이 그러하다.

쿼드트리는 네 개의 자식 노드를 가지는 데이터 구조로써, 이 차원 공간을 재귀적으로 사분면으로 분할하는 응용에 자주 사용된다. 쿼드트리는 같은 값을 가지는 자식 노드들을 통합하는 것으로 데이터를 압축하는 기법으로, 하위 노드들이 같은 값을 가질수록 영상을 더 많이 압축한다(그림 5). 본 논문은 쿼드트리를 사용하여 mvd와 블록 타입 심벌(mb_type과 sub_type)에서의 중복을 제거하는 구조를 제공하며, Motion Vector Tree (MVT)라고 부른다. 블록 타입은 제안하는 타입 코드로 표현하는데, 그림 6은 제안하는 타입 코드를 보여주고 있으며 대문자는 타입 코드를, 소문자는 mvd나 mvd의 대표 값(dominant value)을 나타낸다. mvd에 00 00 00, 00 00 00 01 이

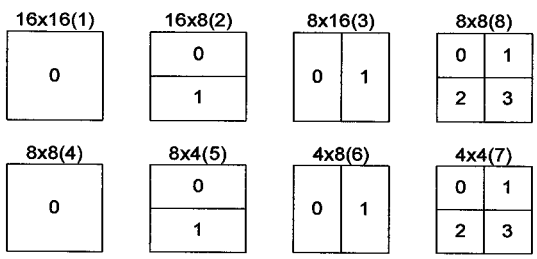


그림 4. H.264의 매크로 블록 타입

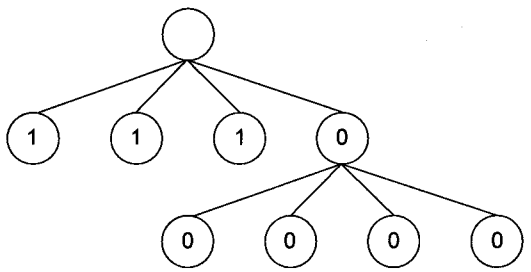


그림 5. 쿼드트리

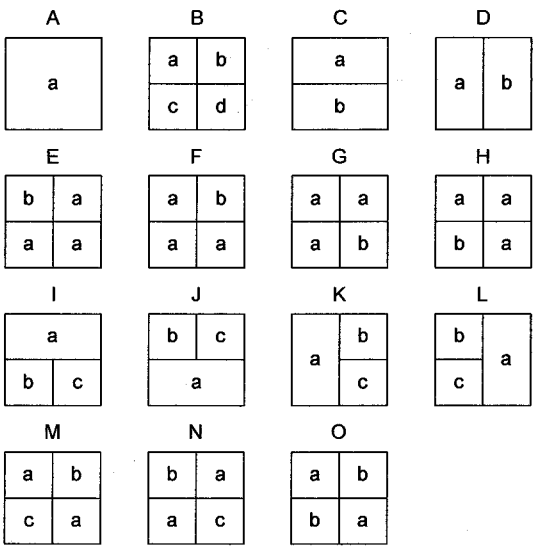


그림 6. 제안하는 타입 코드들 : 대문자는 타입 부호를 나타낸다. 소문자는 mvd와 mvd의 대표 값을 나타낸다.

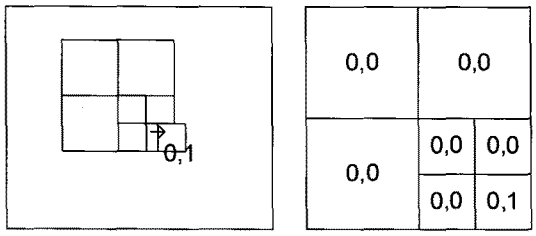


그림 7. 서브-블록에서의 움직임 추정 예

있다고 가정하면(그림 7), 이들 코드들은 8 4447 00 00 00, 00 00 00 01 로 H.264의 헤더에 나타난다. 이 코드는 매크로 블록 타입은 8, 서브 매크로 타입은 4447, x축의 mvd 성분은 0 0 0, 0 0 0 0으로, y축의 mvd 성분은 0 0 0, 0 0 0 1을 나타낸다. MVT에서는 이것을 A0 G0 G01 로 나타낸다. A0 는 x축에서의 타입 코드(A)와 mvd 값(0)이며, G0 G01 는 y축에서의 타입 코드와 mvd 값들이다. 그림 6에서와 같이 G는 두 개의 값들을 요구하며, 이들은 대표 값(0)과 나머지 값이다. 이때 하위 노드가 다시 나누어지면, 하위 노드 값이 나머지 값 대신 주어진다(G01 : 0이 대표 값이며, 1이 나머지 값이다). 위의 예에서 19개의 심벌들은 7개의 심벌들로 줄어들며, 63%의 감소가 이루어지는 것을 볼 수 있다.

제안하는 알고리즘에서 타입 코드와 mvd의 비트를 포함하는 헤더 비트 율은 아래와 같이 계산되어진다.

$$R_{Header} = R_{Type} + R_{Mvd} \tag{3}$$

헤더의 비트 율은 mb_type과 최대 네 개의 sub_type로 이루어진 R_{Type}과 최대 16개의 mvd를 가지는 R_{Mvd}로 이루어진다. 기본적으로 MVT는 H.264의 헤더와 같은 개수의 부호를 가진다. 그러나 8 4444 0001과 같은 경우는 MVT가 좋은 성능을 제공해 준다. 즉, 4444와 0001의 중복을 제거하여 부호화기의 성능을 향상시키며, 이것은 초저비트율 부호화에서는 중요한 요소이다.

그림 8은 제안하는 알고리즘의 플로우차트를 보여준다. 매크로 블록에서 RDO 계산 결과에 따라 파티션을 나누고 첫 번째 파티션 정보를 읽어 들인다. 파티션의 타입 코드를 헤더에 저장하고 타입 코드에 따라 필요한 대표 mvd값과 나머지 값을 계산한 다음, 계산된 값들을 저장한다. 다음 파티션 정보를 읽어 들이고 위의 수행을 반복한다.

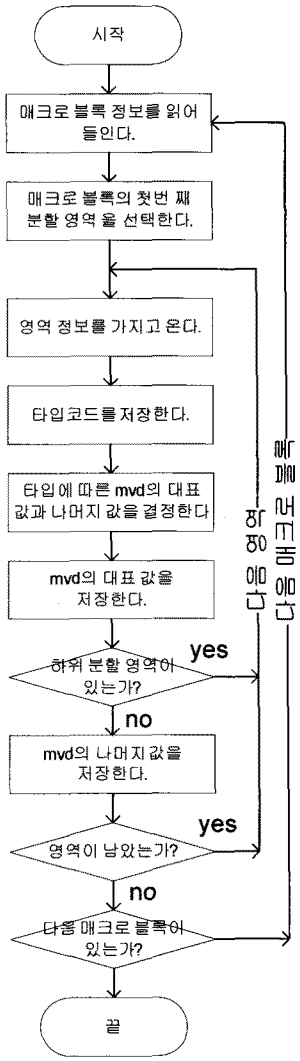


그림 8. 제안하는 알고리즘의 플로우차트

4. 실 험

제안하는 알고리즘의 효용성을 위한 실험은 Joint Video Team(JVT)에서 제공하는 JM12.4 software를 사용했으며, 선택된 영상 열들은 QCIF(176×144) 해상도를 가진다. 다섯 개의 영상 열 "Foreman," "Akiyo," "Mobile," "Mother_daughter,"와 "Stefan" 영상들이 실험에서 사용되었다. 사용 프로파일은 High 프로파일(ProfileIDC=100)을 사용하였다. QP는 I : 28, P : 28 이고 움직임 추정에서 최대 검색 범위는 16이고 움직임 벡터 해상도는 1/4 픽셀이다. Context adaptive binary arithmetic coding

(CABAC)와 Hadamard 변환을 사용하였다. 프레임 율은 30 fps이며, 영상 부호화 구조는 IPPP이고, 각 영상 수는 100장이다. 실험 결과에서의 감소율은 헤더와 전체 비트열의 비트 율 감소를 보여준다. 표 2에서 Bitrate(%)는 비트 율 변화에서의 백분율을 의미하고 양의 값은 증가를, 음의 값은 감소를 의미한다. 제안하는 알고리즘의 성능은 수식 (4)로 테스트된다.

$$\Delta Bitrate = (Bitrate_{JM} - Bitrate_{MVT}) / Bitrate_{JM} \quad (4)$$

JM software와 MVT의 비트 율 비교는 표 2에서 4와 그림 9 에 표시하였고, 이 실험들에서 MVT의 엔트로피 부호화는 huffman coding이 사용되었다. 이 실험의 결과에서 MVT는 QP가 28일 때, 평균적으로 헤더 비트 열에서는 74.85%를 감소시켰으며, 전체 비트 열에서는 24.86%를 감소시켰다. 전체 비트 열에서 QP가 32일 때는 32.51%, 22일 때는 14.48% 감소시켜, 최대 32.51%의 비트율 감소를 보여주고 있다. 표 2-4에서 알 수 있듯이, Mobile과

표 2. JM software와 MVT 결과 비교 : 각 100장, QP(I : 28, P : 28)

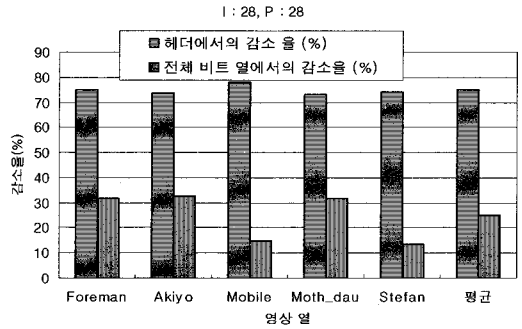
	JM12.4 (Exp-Golomb) (바이트)	MVT (바이트)	헤더에서의 감소 율 (%)	전체 비트 열에서의 감소율(%)
Foreman	21,364	5,312	75.14	31.68
Akiyo	4,342	1,186	73.68	32.57
Mobile	33,249	7,285	78.09	14.85
Moth_dau	7,334	1,979	73.02	31.54
Stefan	27,643	7,096	74.33	13.67
평균			74.85	24.86

표 3. JM software와 MVT 결과 비교 : 각 100장, QP(I : 28, P : 32)

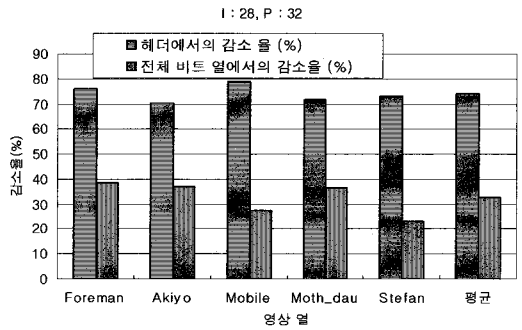
	JM12.4 (Exp-Golomb) (바이트)	MVT (바이트)	헤더에서의 감소 율 (%)	전체 비트 열에서의 감소율 (%)
Foreman	15,197	3,665	75.88	38.69
Akiyo	2,750	811	70.50	37.06
Mobile	26,506	5,566	79.00	27.31
Moth_dau	4,646	1,309	71.82	36.34
Stefan	24,082	6,466	73.15	23.13
평균			74.07	32.51

표 4. JM software와 MVT 결과 비교 : 각 100장, QP(I : 28, P : 22)

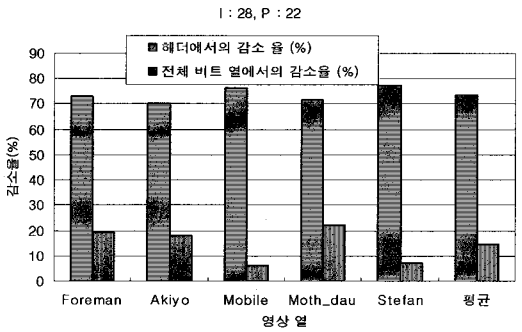
	JM12.4 (Exp-Golomb) (바이트)	MVT (바이트)	헤더에서의 감소 율 (%)	전체 비트 열에서의 감소율 (%)
Foreman	31,432	8,410	73.24	19.12
Akiyo	6,957	2,086	70.02	17.85
Mobile	36,185	8,497	76.52	6.26
Moth_dau	13,850	3,928	71.64	22.11
Stefan	31,031	7,086	77.17	7.04
평균			73.72	14.48



(a)



(b)



(c)

그림 9. 비트 감소율 결과 그래프

Stefan 영상 열처럼 복잡한 영상에서는 헤더의 비율이 전체 비트 열에서 작기 때문에 전체 비트 열에서의 감소율이 작게 나온다. 기존 방식의 Exp-Golomb 방식에서는 15%.의 압축 성능을 보이고 있다[2]. MVT 처리를 위한 시간은 무시할만하다, 왜냐하면 모드결정은 RDO에서 결정되고 대부분의 시간이 RDO에서 소모되기 때문이다. 우리는 결과에서 H.264의 헤더 압축을 위한 방법으로 MVT가 효율적이라는 것을 알 수 있었다.

5. 결 론

본 논문은 H.264의 헤더를 압축하기 위해 타입 코드와 쿼드트리를 이용한 새로운 구조를 제안하며, 이 구조를 사용하여 움직임 정보와 모드 정보를 압축하는 것을 목적으로 한다. 새로운 압축 표준안인 H.264는 보다 좋은 압축 성능을 제공하고 있지만 채택하고 있는 기술들로 인해, 오히려 헤더가 전체 비트열의 50%까지 차지한다. 헤더에서 세 가지 중복성이 있다는 사실을 기반으로 제안된 알고리즘에서는 중복되는 mvd와 블록 타입을 효과적으로 표현한다. 실험에서 제안하는 알고리즘이 JM12.4 보다 최대 32.51%의 비트 감소를 제공하는 것을 보여주고 있다.

참 고 문 헌

[1] ISO/IEC 14496-10 and ITU-T Rec. "Draft ITU-T Rec. and FDIS of Joint Video Spec. (H.264 | ISO/IEC 14496-10 AVC)," *JVT of VCEG and MPEG*, Doc. JVT-G050r1, May 2003.

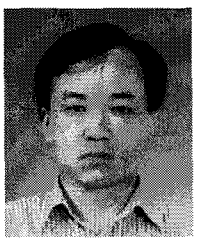
[2] 백성학, 문용호, 김재호, "다중 부호어를 이용한 효율적인 H .264/AVC 동적 부호화 방법," 한국통신학회논문지, 제29권, 8C호, pp. 1055-1061, 2004.

[3] W. Di, G. Wen, H. Mingzeng, and J. Zhenzhou, "An Exp-Golomb encoder and decoder architecture for JVT/AVS," *ASIC, Proc. 5th International Conference*, Vol.2, pp. 910-913, 21-24 Oct. 2003.

[4] D.-K. Kwon, M.-Y. Shen, and C.-C. J. Kuo, "Rate Control for H.264 Video With Enhanced Rate and Distortion Models," *IEEE Trans. Circuits and Systems for Video Technology*, Vol.17, Issue 5, pp. 517-529, May 2007.

[5] D.-K. Kwon, M.-Y. Shen, and C.-C. J. Kuo, "A Novel Two-Stage Rate Control Scheme for H.264," *Multimedia and Expo, 2006 IEEE International Conference*, pp. 673-676, July 2006.

[6] C.-W. Wong, O. C. Au, and R.C.-W. Wong, "Advanced Macro-block Entropy Coding in H.264," *Proc. ICASSP, Acoustics Speech and Signal Processing*, Vol.1, pp. I-1169-I-1172, 15-20 Apr. 2007.



이 동 식

1996년 경북대학교 (공학사 - 전자공학) 졸업

1997년 3월~1999년8월 경북대학교 (공학석사 - 전자공학)

2000년 3월~현재 경북대학교 전자공학과 박사

관심분야 : 영상처리, 영상압축, 영상전송

김 영 모

1980년 경북대학교 (공학사 - 전자공학) 졸업

1980년 3월~1983년2월 한국과학기술원 (공학석사 - 전자공학)

1983년 3월~1989년2월 한국과학기술원 (공학박사 - 전자공학)

1997년~현재 경북대학교 공과대학 교수

관심분야 : 컴퓨터 그래픽스, 멀티미디어, 비쥬얼 컴퓨팅, 영상처리