Journal of Korea Multimedia Society Vol. 13, No. 10, October 2010(pp. 1487-1493)

단말 모드 방식을 도입한 H.264의 움직임 벡터 압축

이동식[†], 김영모[††]

요 약

H.264는 기존의 표준안들보다 좋은 성능을 내기 위해서 더 세밀하고, 더 많은 움직임 정보를 처리한다. 하지만, 움직임 벡터가 기존 표준안에 비해 높은 비율을 차지하게 되어, 이에 대한 고려가 필요하게 되었다. 본 논문에서는 움직임 벡터를 효율적으로 처리하기 위해 단말 모드를 사용하는 새로운 방법을 제안한다. 제안하는 알고리즘은 단말 모드를 이용하여 모드들의 분포를 집중시키고, 더 적은 모드 종류를 이용하여 움직임 벡터의 압축을 수행한다. 본 논문에서 제시하는 알고리즘은 현재의 4×4 움직임 벡터 행렬 뿐만 아니라 8×8 행렬 등으로 확장시킬 수 있다. 제안하는 알고리즘은 헤더에서 12.68%, 전체 비트스트림에서 9.7%의 감소율을 보여준다.

Motion vector compression in H.264 with leaf mode

Dong-Shik Lee[†], Young-Mo Kim[††]

ABSTRACT

H.264 processes more detailed and more motion information for the compression efficiency. However, motion vector of H.264 takes more portion than previous standards such as MPEG-1/2/4 do, so that it is needed to consider motion vectors. This paper proposes the new algorithm with leaf mode in order to compress the motion vector efficiently. The proposed algorithm concentrates modes' distribution with leaf mode and carries out the compression of motion vector with less modes. The proposed algorithm adopted in current 4×4 motion vector matrix also can be extend to 8×8. The experiments shows that the proposed algorithm reduces up to 12.68% at header and 9.7% at resultant bitstream.

Key words: H.264, Motion vector(움직임 벡터), Quadtree(쿼드트리), Leaf mode(단말 모드)

1. 서 론

최근에 제정된 H.264[1,2]는 최대의 압축률을 이루기 위해, 새로운 기술을 채택하거나 기존 기술들에 더 세밀하고 많은 처리를 요구한다. 새로운 기술들은 정수 변환, 디블로킹 필터, 다중 참조 프레임, 효율적인 엔트로피 부호화 등이다. 이중에서 다중 참조 프레임 처리와 세밀해진 움직임 추정/보상으로 압축률이 더욱 향상되었다. 또한 율-왜곡 최적화(rate-distortion optimization) 기법을 통해 움직임 보상을 할 때, 왜곡 뿐만 아니라 비트율을 고려하기 때문에 가장 효율적인 압축 성능을 보장해 준다. 율 조절은 특히 협 대역 동영상 통신에서 중요한 역할을 한다.

멀티미디어 압축과 전송 분야에서 널리 사용되고 있는 MPEG-2가 발표된 이후 움직임 추정/보상(Motion Estimation/Compensation)과 후 처리를 위한 블록 크기는 작아지고 세밀해지는 경향인데, 예를 들어 MPEG-2는 16×16 블록을 사용하고 MPEG-4

※ 교신저자(Corresponding Author) : 이동식, 주소 : 대구광역시 북구 복현동 경북대학교 모바일 테크노 빌딩 405호 디아이랩(702-701), 전화 : 053)959-5541 FAX : 0505)959-5541, E-mail : lds@dilab.co.kr
접수일 : 2010년 4월 30일, 수정일 : 2010년 7월 19일
완료일 : 2010년 9월 1일

[†] 준회원, 디아이랩
[††] 준회원, 경북대학교 IT대학 전자공학부 교수
(E-mail : ymkim@ee.knu.ac.kr)
※ 본 연구는 과학기술부의 재원으로 한국연구재단의 지원을 받아 수행된 이공분야 일반연구자지원 사업의 연구결과입니다.

는 16×16과 8×8 블록을 사용할 수 있다. 이들 표준안들에서는 움직임 추정/보상으로 발생하는 움직임 벡터 차이(Motion Vector Difference : mvd)를 포함하는 헤더(Header)의 크기가 전체 비트열에 비해 무시할 만 수준이기 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 블록까지 사용하고 다중 참조 영상을 사용하기 때문에 더 정밀한 움직임 보상을 수행한다. 만약 H.264에서 선택된 모드가 4×4 블록이라면 최대 16개의 움직임 벡터와 4개의 참조 영상 색인들이 전송된다. 더욱이 H.264에서 채택하고 있는 모드 선택 기법으로 인해 매크로 블록(Macroblock)과 서브-매크로 블록(Sub-Macroblock) 타입까지 결과 비트열에 전송하며, 이런 기술들로 인해 헤더가 전체 비트열의 50%까지 차지하기도 한다. 하지만 H.264의 기술들은 주로 잔여 데이터의 압축에 주안점을 두고 있으며, 이전 표준안에 비해 헤더가 중요해졌지만 여전히 크게 고려되고 있지 않다. H.264에서는 지수 골롬(Exp-Golomb) 부호화 방식을 사용하는 가변 길이 부호화를 사용하여 잔여 데이터를 제외한 나머지 구문 요소를 부호화하고 있다[3,4].

H.264의 움직임 벡터와 헤더를 압축하기 위한 여러 연구가 있었지만 이들 연구들은 모드 선택에서 움직임 벡터의 개수나 비트율을 줄이는 방향으로 접근을 하고 있다. Kwon [5,6]등은 연구에서 헤더 정보를 나타내는 모델을 제안하였는데, 이 모델은 '0'이 아닌 움직임 벡터 요소와 움직임 벡터의 개수에 대한 함수로 표현한다. Wong [7]등은 H.264에서 헤더를 압축하기 위해서 헤더의 크기를 고려하는 엔트로피 부호화 방식을 제안하였으며, 결과에서 10%의 비트열 압축이나 같은 비트율에서 0.3~0.4 dB 향상을 보여주었다.

H.264에서는 나무 구조 움직임 추정 방식(Tree Structure Method : TSM)이라는 방식을 사용하여 4개의 모양과 크기 정보를 이용하여 가변 블록 크기 움직임 추정/보상을 해 주고 있다. 하지만 이 방식은 제한된 모양만을 나타낼 수 있으며, 크기 정보를 가져야 하기 때문에 같은 모양임에도 더 많은 비트율을 요구하게 되며, 계층적인 구조를 사용할 수 없다. 본 논문에서는 기존 방식이 모양 정보와 크기 정보를 함께 나타내는 나무 구조 방식의 문제점을 단말노드로 해결하는 단말 모드 방식(Leaf Mode Method : LMM) 방식을 제안한다. 제안하는 방식은 기본적으로 대표 움직임 벡터를 전송함으로써 움직임 벡터의 압축 효과를 가져오고 있으며, 단말 모드로 인해 크기 정보를 전송할 필요없는 계층적인 구조를 제공해 준다.

본 논문의 나머지 부분은 다음과 같이 구성된다. 2절에서는 H.264에서 사용되는 움직임 추정/보상에 대해 알아보고, 3절에서는 H.264의 헤더에 존재하는 블록 형태와 움직임 벡터 전송 방식과 제안하는 두 가지 방식을 설명한다. 4절에서는 기존 방식과 제안하는 방식들 간의 비교 분석한 실험 결과가 주어지며 제안하는 알고리즘의 성능을 보여준다. 5절에서 최종적인 결론을 맺는다.

2. H.264의 움직임 벡터

MPEG-2/4에서는 고정 크기 블록 움직임 추정/보상 방식을 사용한다. H.264에서는 가변 블록 크기 움직임 추정/보상 방식을 사용하기 때문에, 이전 표준안의 고정 크기 블록 움직임 추정/보상 방식보다 더 뛰어난 압축 성능을 보장해 준다. 또한, 이전 표준안에 비해 움직임을 더 세밀하게 처리하기 위해 독립적으로 처리되는 블록의 크기가 16×16에서 4×4까지 작아졌다.

하지만, 가변 블록 크기 움직임 추정/보상 방식은 움직임 벡터의 위치나 크기 정보 등을 추가로 전송을 해 주어야만 한다. 이 문제로 인해, 이전 표준안에서는 고정 크기 블록 움직임 방식을 사용하였다. 이를 보상하기 위해 H.264는 나무 구조 움직임 추정/보상 방식을 사용하여, 4개의 모양으로 매크로 블록 내의 움직임 벡터를 표현하고 전송하고 있다. 이런 방식을 취함으로써, 어느 정도 움직임 벡터를 표현하고 있지만, 여전히 개선의 여지가 존재하고 있다.

본 절에서는 H.264 에서 움직임 벡터를 처리하는 과정을 살펴본다. 움직임 벡터는 이전 영상과의 차이를 블록 단위로 처리하는 것이며, 참조 프레임과 현재 프레임간의 가장 작은 오류를 가지는 매크로 블록의 위치를 지정하는 것이다. 현재 표준안들은 더 나아가 주변 움직임 벡터와의 차이만을 전송하고 있

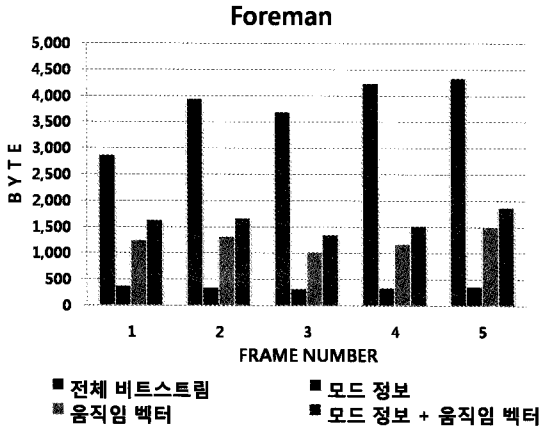

그림 1. Foreman 분석

다. 잔여 데이터의 압축 기술이 발전함에 따라 전송되어지는 움직임 벡터는 이전 표준안에 비해 더욱 많아지게 되었다. 그림 1은 Foreman 영상에서의 비트율을 보여주고 있다. 모드 정보와 움직임 벡터를 합친 량이 전체 비트율에서 45%가 됨을 볼 수 있다. 앞서에서도 얘기했듯이, H.264에서는 가변 블록 크기 움직임 방식 채택함과 동시에 움직임 정보량이 많아지게 되었다.

새로운 기술들 중에서 H.264는 율-왜곡 최적화이라는 개념을 사용하여 왜곡과 비트율간의 조정을 통해 압축율과 화질 사이의 균형을 이루고 있다. 아래 수식 1이 율-왜곡 최적화를 설명하고 있다.

$$J = D + \lambda \cdot \sum R \quad (1)$$

R은 비트율을 나타내고 있으며, 여기에는 잔여 데이터와 헤더 데이터를 포함하고 있다. D는 왜곡을 나타내며, J는 비용함수를, 그리고 λ는 라그랑지안 곱수를 나타낸다. 이전 표준안에 비해 커진 비트율을 가지는 헤더는 움직임 벡터와 모드 정보가 많은 부분을 차지한다. 그러므로 본 논문에서는 움직임 벡터와 모드 정보를 움직임 정보로 사용한다.

3. 제안하는 움직임 벡터 압축 알고리즘

기존 방식의 나무 구조 방식은 같은 값들이 분포하더라도, 이를 처리할 수 있는 방법이 없었으며, 계층적인 구조가 불가능하다. 본 논문에서는 쿼드트리를 이용한 움직임 벡터 압축 알고리즘을 제안하며, 본 절에서 제안하는 알고리즘을 설명한다.

3.1 움직임 벡터의 대표 값

기존 방식에서는 대표 값이라는 개념이 없기 때문에, 전송 블록에서 같은 움직임 벡터가 발생하더라도 모든 움직임 벡터들을 전송해야 한다. 본 논문에서는 2×2블록을 기본 구조로 사용하는 단말 모드 방식을 제안한다. 기본 구조는 2×2블록에서 정의된 모양에서 가장 많은 블록을 차지하거나 주어진 위치의 움직임 벡터를 대표 움직임 벡터로 선정하고, 대표 값을 전송함으로써 같은 블록내의 같은 움직임 벡터는 전송하지 않는다. 움직임 벡터 대표 값을 사용하기 위해서는 대표 움직임 벡터를 상위 블록들로 전송하고, 다시 상위 블록 레벨에서 대표 움직임 벡터를 선정하는 방식을 사용한다.

대표 값 선정으로 주어진 모양에서 대표 값과 같은 값은 전송하지 않게 된다. 예를 들어, 그림 2에서와 같은 4×4 움직임 벡터 행렬이 존재한다면, 기존 방식에서는 01, 23, 01, 23를 전송한다.

대표 값을 선정하게 되면, 왼쪽 위 2×2 블록과 아래 2×2 블록에서 대표 값 0와 나머지 값 1, 오른쪽 위 2×2 블록과 아래 2×2 블록에서 2와 나머지 값이 3이 된다. 전체 블록(4×4)에서 0이 대표 값이 되고, 2가 나머지 값이 된다. 그러므로 대표 값 방식으로 전송하면 02, 13만을 전송한다. 가장 나쁜 경우인, 모든 움직임 벡터가 서로 다른 경우에는 기존 방식과 같은 움직임 벡터를 전송하게 된다. 하지만, 이러한 구조는 계층적인 구조를 가져야만 가능한 구조로써, 제안하는 방식인 LMM은 이러한 계층적인 구조를 지원한다.

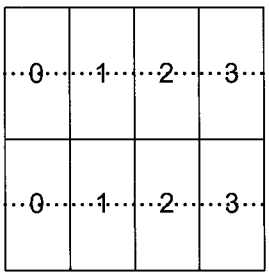

그림 2. 4×4 블록에서 움직임 벡터의 대표 값이 있는 예

3.2 단말모드를 이용한 압축 방식

제안하는 방식은 단말 모드를 사용하여 계층적인 구조가 가능해지며, 크기 정보를 위해서 사용하는 기

존 방식의 나머지 모드들을 사용하지 않는다. 기존 방식에서는 4×4 블록과 2×2 블록을 구분하여 8개의 모드 정보를 사용하고 있지만 구분되는 모드의 종류는 4개 뿐이다(그림 3).

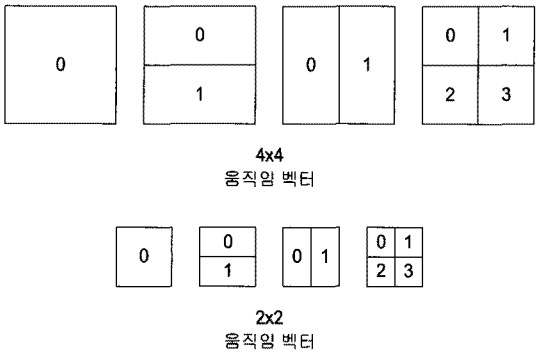

그림 3. 나무구조 움직임 모드 정보

본 논문에서 제안하는 방식은 8개의 모드가 아닌 4개의 구조와 한개의 단말 모드를 가진 구조를 제안한다(그림 4). 8개의 심볼보다 5개의 심볼에서 발생하는 비트수가 감소하고, 움직임 벡터의 대표 값을 전송함으로써 전송되는 움직임 벡터의 개수를 줄임으로써, 전체 움직임 정보에 소요되는 비트 수를 줄인다.

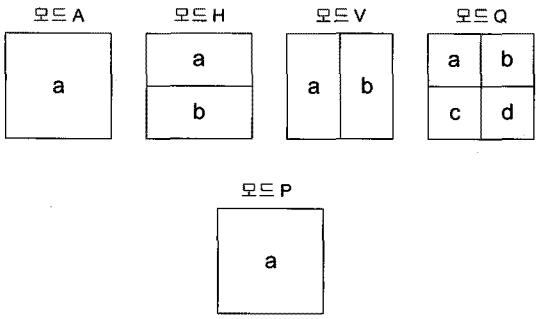

그림 4. 단말 모드를 가진 방식

제안하는 알고리즘에서 추가되는 5번째 모드(P 모드)는 마지막 노드임을 가리켜서 모드 정보의 끝을 나타내 줌으로써 정확한 복호가 가능하다. 이러한 방식은 기존의 방식의 모든 구조를 표현하면서도 심볼의 개수를 줄여주고 있다. 특히, 특정 모드(P 모드)의 빈도 수를 높여 주어 엔트로피 부호화기의 성능을 높여준다.

그림 4에서 소문자 a~d는 해당 블록의 움직임 벡터이고, a는 움직임 벡터의 대표 값이다. 모드 P(단말 모드)는 하위 레벨의 모든 블록들의 움직임 벡터가 같은 모드이며 더 이상 분할되지 않는다. 모드 A는 모든 P가 아니면서 하위 레벨들의 블록들의 대표 값이 같은 경우이다. 모드 H(Horizontal)은 두개의 상위 블록들의 대표 값이 같고, 두개의 하위 블록들의 대표 값이 같으면서 상위 블록들의 대표 값과 하위 블록들의 대표 값이 다른 경우이다. 모드 V(Vertical)은 두개의 왼쪽 블록들의 대표 값이 같고, 두개의 오른쪽 블록들의 대표 값이 같으면서 왼쪽 블록들의 대표 값과 오른쪽 블록들의 대표 값이 다른 경우이다. 모드 Q(quad)의 경우는 모든 하위 블록들의 대표 값이 다른 경우이다. 모드 A와 P의 같은 점은 하위 블록들의 대표 값이 같은 것이고, 다른 점은 모드 A의 모든 하위 블록들의 움직임 벡터가 값이 같지 않다는 것이다.

단말 모드 방식은 크기를 정의하지 않으며 계층적인 구조로 확장할 수 있으므로, 현재의 구조인 4×4 블록 크기가 아닌 8×8 블록의 경우에도 그대로 적용할 수 있다. 블록을 계층적으로 분할하다가 단말 모드인 P를 만나거나 마지막 레벨이면, 현재 블록의 분할을 멈추고 다음 블록을 다시 분할한다.

단말 모드 방식에서 발생하는 모드의 비트량을 확인 해 보기 위해 허프만 테이블을 생성하고 비트량을 계산해 본다. 동영상 실험에서 자주 사용되는 11장의 영상에서 계산된 각 모드들의 빈도수는 그림 5에서 보여주고 있다. 모드 P가 가장 많은 빈도를 가지고 있어, 발생 빈도가 단말 모드인 P 모드에 모이는 경향을 보여주고 있다. 소요되는 비트량을 계산하기 위해 그림 5의 빈도수를 가지고 계산한 허프만 테이블은 표 1과 같다. 허프만 테이블과 그림 1이 빈도수를 가지고 계산을 해 보면 20,712비트이다. 비교를 위해 기존 방식의 비트량을 계산해 보기 위해 그림 6의

표 1. LMM 방식들의 모드들의 비트수

모드	비트 수	이진수
A	4	0000
H	3	001
V	4	0001
Q	2	01
P	1	1

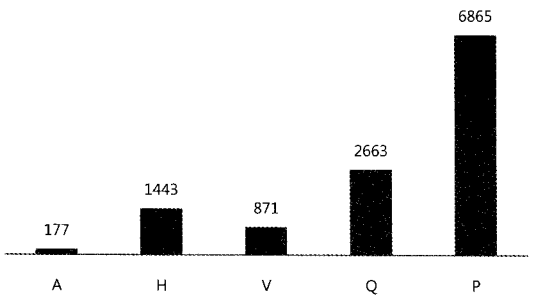

그림 5. LMM 방식의 모드들의 빈도수

빈도수로 계산된 기존 방식의 허프만 테이블을 보여준다(표 2). 허프만 테이블과 빈도수를 가지고 계산을 해 보면 28,277 비트이다. 단말 모드 방식이 기존 방식에 비해 7,565비트가 감소하며 매크로 블록이 4,356개(11장, 1장당 396개)라면 1.73비트가 감소하고, 26.75%의 감소율을 보여준다. 위에서 서술한 내용에서는 기존 방식에서 사용하는 심볼의 개수를 줄이고 한 모드의 빈도수가 커짐으로써, 모드 정보에서 사용되는 비트량을 줄이는 근거를 제시해주고 있다.

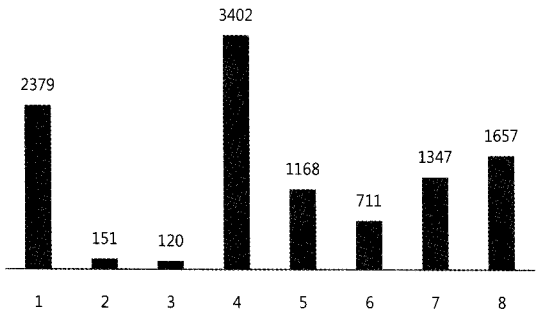

그림 6. TSM의 모드들의 빈도수

표 2. TSM 방식들의 모드들의 비트수

모드	비트 수	이진수
1	2	01
2	5	00001
3	5	00000
4	2	11
5	3	001
6	4	0001
7	3	100
8	3	101

4. 결 과

기존 방식들과 제안하는 방식은 움직임 추정에서 결정되어진 움직임 벡터 정보를 무손실로 압축한다. 따라서 제안 방식에 의하여 동영상의 화질은 저하되지 않으며 기존 방식들을 적용하였을 경우와 동일한 복원 영상을 얻게 된다. 제안하는 알고리즘의 효용성을 위한 실험은 Joint Video Team(JVT)에서 제공하는 JM16.1 software를 사용했으며, 선택된 영상들은 CIF(352×288) 해상도를 가진다. 5개의 영상 "Mobile", "Bridge", "Bus", "City"와 "Football" 등의 영상들이 실험에서 사용되었다. 사용 프로파일은 High 프로파일(ProfileIDC=100)을 사용하였고, 표 3은 기존의 H.264/AVC JM16.1의 방법과 성능 비교를 위한 실험 파라미터들을 보여주고 있다.

실험 결과에서의 감소율은 헤더와 전체 비트열의 비트율 감소를 보여준다. 표 4~8에서 Bitrate(%)는 비트율 변화에서의 백분율을 의미하고 양의 값은 증가를, 음의 값은 감소를 의미한다. 제안하는 알고리즘의 성능은 수식 3으로 테스트된다.

$$\triangle Bitrate = \frac{Bitrate_{JM} - Bitrate_{LMM}}{Bitrate_{JM}} \quad (3)$$

JM software와 제안하는 방식과의 비트율 비교는 표 4~8에 나타내었고, 이 실험들에서 LMM의 엔트로피 부호화는 허프만 부호화가 사용되었다. 허프만 부호화에서는 통계 테이블을 사용한 방법과 사용하지 않는 방법을 사용하였다. 통계적 허프만 테이블을 사용하는 경우, 평균적으로 헤더 비트열에서 LMM은 12.68% 감소시켰으며, 통계적 허프만 테이블을

표 3. 실험에 사용한 파라미터들

Parameters	Settings
Sequences	CIF(352×288)
Searching range	16
Reference frame	5
Qp	28
Sequence type	IPPP
Entropy coding	CABAC
GOP	16
Search range restrictions	None
Frame rate	30 fps
Block type	All used

표 4. JM software와 제안하는 방식들의 결과 비교 : Mobile

	모드/통계(바이트)	MV(바이트)	모드+MV(바이트)	감소율(%)
TSM	308/324	497	805/821	-
LMM	242/245	475	717/720	10.9/12.3

표 5. JM software와 제안하는 방식들의 결과 비교 : Bridge

	모드/통계(바이트)	MV(바이트)	모드+MV(바이트)	감소율(%)
TSM	121/154	220	341/374	-
LMM	98/99	217	315/316	7.6/15.5

표 6. JM software와 제안하는 방식들의 결과 비교 : Bus

	모드/통계(바이트)	MV(바이트)	모드+MV(바이트)	감소율(%)
TSM	402/425	1,274	1,676/1,699	-
LMM	321/321	1,192	1,513/1,513	9.7/11.0

표 7. JM software와 제안하는 방식들의 결과 비교 : City

	모드/통계(바이트)	MV(바이트)	모드+MV(바이트)	감소율(%)
TSM	171/193	278	449/471	-
LMM	135/142	271	406/413	9.6/12.3

표 8. JM software와 제안하는 방식들의 결과 비교 : Football

	모드/통계(바이트)	MV(바이트)	모드+MV(바이트)	감소율(%)
TSM	435/467	1,297	1,732/1,764	-
LMM	361/361	1,186	1,547/1,547	10.7/12.3

사용하지 않는 경우, 평균적으로 LMM은 9.7% 감소시켰다. 제안하는 방식들을 처리하기 위한 시간은 무시할만하다, 왜냐하면 모드결정은 RDO에서 결정되고 부호화기에서 소모되는 대부분의 시간이 RDO에서 소모되기 때문이다.

5. 결 론

국제 동영상 표준안들이 압축률을 향상시키기 위해 많은 노력을 해 왔으며, 특히 왜곡과 비트율을 동시에 고려하는 RDO 개념을 도입한 H.264는 최신 압축 표준안으로 이전 표준안인 MPEG-2에 비해 50%의 압축 향상을 이루고 있다. 또한, 움직임을 효율적으로 처리하기 위해 새로운 기술들을 도입하였기 때문에, 움직임을 처리하기 위한 헤더 정보의 증가를 가져왔으나, 해당 표준안에서는 여전히 잔여 데이터를 압축하기 위한 기술에 집중하였다. 그러므로, 헤더내의 움직임 벡터 성분이 이전 표준안에 비해 월등하게 커지고, 여기에 대한 고려가 요구되고 있다.

본 논문에서는 움직임 벡터가 가지고 있는 특징들을 분석하여 효율적인 움직임 정보 표현 방식을 제안한다. 제안하는 방식은 단말 모드를 추가하여 매크로블록 내 움직임 벡터의 대표 값을 전송하여 같은 움직임 벡터를 전송하지 않으며, 단말 모드의 빈도수가 커지게 된다. 또한, 기본 구조를 2x2 블록을 기본 구조로 확장가능하다. 기존 방식에서는 모양과 크기 정보만을 가지는 구조로 이루어져 있지만, 단말 정보를 가지는 모드를 추가함으로써, 크기 정보를 가지는 모드가 필요 없게 됨과 동시에 계층적인 구조를 가지게 된다. 제안하는 알고리즘의 가장 큰 특징은 대표 값을 전송할 수 있게 함으로써, 대표 값과 같은 값을 가지는 값을 전송할 필요가 없게 하는 것이다. 기존 방식은 대표 값 개념이 없이 단순한 모양 정보만을 가지고, 해당하는 모양 정보로 움직임 벡터를 전송하

고 있다. 본 논문의 결과에서 보면, 테스트 영상들에서 헤더의 압축 성능이 12.68% 향상되었음을 알 수 있다.

참 고 문 헌

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology,* Vol. 13, No. 7, pp. 560-576, July 2003.

[2] S. D. Kim and J. B. Ra, "An efficient motion vector coding scheme based on minimum bitrate prediction," *Image Processing, IEEE Transactions on,* Vol. 8, No. 8, pp. 1117-1120, August 2002.

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

[4] 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.

[5] D.-K. Kwon, M.-Y. Shen, and C. C. J. Kuo, "Rate control for h.264 video with enhanced rate and distortion models," *IEEE Transactions on Circuits and Systems for Video Technology,* Vol. 17, No. 5, pp. 517-529, May 2007.

[6] 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, 9-12 July 2006.

[7] C.-W. Wong, O. Au, and R.-W. Wong, "Advanced macro-block entropy coding in h.264," Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on, Vol. 1, pp. I-1169-I-1172, 15-20 April 2007.

이 동 식

1996년 경북대학교 (공학사-전자공학) 졸업
1997년 3월~1999년 8월 경북대학교(공학석사-전자공학)
2000년 3월~2010년 2월 경북대학교(공학박사-전자공학)
2010년~현재 디아이랩 재직
관심분야 : 영상처리, 영상압축, 영상전송

김 영 모

1980년 경북대학교(공학사-전자공학) 졸업
1980년 3월~1983년 2월 한국과학기술원(공학석사-전자공학)
1983년 3월~1989년 2월 한국과학기술원(공학박사-전자공학)
1997년~현재 경북대학교 공과대학 교수
관심분야 : 컴퓨터 그래픽스, 멀티미디어, 비쥬얼 컴퓨팅, 영상처리