

MPEG-4 Part 10 동영상 압축 표준 성능 개선을 위한 새로운 부호화 방식

정회원 문 용 호*

A New Coding Method for Improving the Performance of MPEG-4 Part 10 Video Coding Standard

Yong-ho Moon* *Regular Member*

요 약

본 논문에서는 MPEG-4 Part 10 동영상 압축 표준에 대한 새로운 움직임 벡터 부호화 알고리즘을 제안한다. 제안 알고리즘에서는 주변 블록들의 움직임 벡터 분포와 MPEG-4 Part 10의 구문 요소인 매크로블록 유형을 이용하여 주어진 동영상의 움직임 정도를 예측한다. 그리고 예측 결과에 따라서 독립 부호화 방식과 결합 부호화 방식을 적응적으로 선택하여 움직임 벡터 압축을 수행한다. 모의 실험 결과는 부가적인 메모리나 계산량 없이 제안 방식이 기존 방식들보다 우수한 압축 성능을 지님을 보여준다.

Key Words : Motion Vector Difference, MPEG-4 Part 10, 2-D Coding, Exp-Golomb Code, VLC

ABSTRACT

In this paper, we propose a new motion vector coding algorithm suitable for MPEG-4 Part 10 video coding standard. In the proposed algorithm, the amount of the motion for a given video sequence is predicted by using a characteristic of the motion vector distribution for the neighboring blocks as well as the MB_type, which is the syntax element in the standard. And one of the independent coding and the combined coding methods is adaptively employed to compress the motion vector difference. Simulation results show that the proposed algorithm outperforms the conventional methods without additional memory and calculations.

1. 서 론

다양한 멀티미디어 서비스의 출현과 발전은 높은 압축 효율을 전제로 한다. 이러한 요구에 부응하기 위하여 최근에 MPEG-4 Part 10^[1]이 새로운 동영상 압축 표준 기술로 채택되었다. MPEG-4 Part 10 표준에서는 매크로 블록의 다양화, 복수 참조 프레임의 이용, 유효-왜곡 최적화 기법의 도입 등으로 인하

여 압축 성능이 2배 이상 향상되었다. 그러나 기존 압축 표준들과 비교해 볼 때 움직임 벡터 부호화 기법은 크게 개선되지 못하였다.

기존 압축 표준에서와 같이 MPEG-4 Part 10 표준에서도 중간값(Median) 예측에 의하여 움직임 벡터의 가로와 세로 방향 성분들이 추정되어 진다. 그리고 이때 얻어진 각 방향의 움직임 벡터 차(MVD)는 Exp-Golomb 부호어를 이용하여 독립적으로 가

* 이 논문은 2005년도 정부재원 (교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음. (KRF-2005-003-D00255)

* 부산외국어대학교 IT대학 디지털미디어학부 (yhmoon5@pufs.ac.kr)

논문번호 : KICS2006-04-160, 접수일자 : 2006년 4월 5일, 최종논문접수일자 : 2006년 10월 12일

변질이 부호화된다. 이러한 움직임 벡터 부호화 기법은 영상의 복잡도나 움직임 정도와 같은 특징들을 반영하지 못하기 때문에 효과적인 압축이 이루어지지 못한다. 그런데 오늘날 전체 압축 비트량에 있어서 움직임 벡터의 비트량은 무시할 수 없는 큰 비중을 차지하고 있다. 따라서 보다 더 향상된 압축 성능을 얻기 위해서는 움직임 벡터에 대한 효율적인 부호화가 개발되어야 한다.

지금까지 움직임 벡터는 여러 분야에서 다양한 관점으로 연구되어 왔다²⁻⁴⁾. 그 가운데서 압축률 개선에 관한 몇몇 연구들이 시도되었다. Yu⁵⁾는 MPEG-2 압축 표준에 있어서 각 방향 성분의 MVD를 하나로 결합하여 2차원으로 가변길이 부호화하는 방식을 제안하였다. 그러나 이 방식은 중간값 예측을 이용하기 때문에 압축 성능의 개선이 미미하다. 특히, 움직임이 빠르거나 복잡한 영상에 있어서는 성능 저하가 발생한다. 또한 2차원 허프만 부호화를 위한 새로운 통계치 계산이 요구되는 문제를 지니고 있다. MPEG-4 압축 표준에 있어서 중간값 예측의 한계를 극복하기 위하여 최소 비트율 방식⁶⁾이 제안되었다. 이 방식은 영상의 움직임 정도에 따라 움직임 벡터를 효과적으로 예측함으로써 움직임이 많고 복잡한 동영상에서 압축 성능이 향상되었다. 그러나 이 방식은 움직임이 작은 동영상에서 여전히 중간값 예측을 이용하여 독립적인 부호화를 수행하는 한계를 지니고 있다. 더구나 MPEG-4 Part 10 압축 환경에서 우수한 압축 성능을 보이지 못한다. Kimata⁷⁾는 MPEG-4 Part 10 압축 표준에 있어서 중간값 예측에 의하여 얻어진 각 성분의 MVD를 결합하여 가변길이 부호화하는 방식을 제안하였다. 그러나 이 방식에서는 성능이 개선되지 못하였을 뿐만 아니라 도리어 저하되는 경우가 많았다. 이것은 영상의 특성과 MPEG-4 Part 10 표준의 특징을 활용하지 못하였기 때문이다.

본 논문에서는 MPEG-4 Part 10 표준의 압축 성능 향상을 위하여 효과적인 새로운 움직임 벡터 부호화 알고리즘을 제안한다. 영상의 움직임 정도에 관계없이 효과적인 부호화를 수행하기 위하여 본 논문에서는 MVD의 분포를 분석한다. 분석 결과를 토대로 제안 알고리즘에서는 주어진 MVD에 대하여 독립 부호화 방식과 결합 부호화 방식이 선택적으로 적용되어진다. 실험 결과는 부가적인 메모리의 사용 없이 다양한 영상들에 대하여 움직임 벡터 부호화 성능이 향상되어짐을 보여준다.

II. MPEG-4 Part 10에서의 움직임 벡터 부호화 방식

기존 동영상 압축 표준들과 마찬가지로 MPEG-4 Part 10 역시 혼성 부호화 (Hybrid Coding) 기법에 기초하고 있다. 그러나 MPEG-4 Part 10은 다음과 같은 고유한 특징들을 지니고 있다.

- 다양한 매크로 블록 유형에 기초한 계층적 움직임 추정 및 보상.
- 다중 참조 프레임을 이용한 움직임 추정 및 보상.
- 1/4 부화소 단위의 정밀한 움직임 추정.
- 윌-왜곡 최적화 기법을 이용한 성능 개선.
- 새로운 가변길이 부호화 : Exp-Golomb VLC, CAVLC, CABAC.
- 4x4 정수형 DCT 및 In-loop Filtering.

이러한 특징들은 MPEG-4 Part 10에서 2배 이상의 압축 성능 향상을 가져왔다. 특히, 움직임 추정 및 보상과 관련하여 도입된 새로운 방식들은 DCT 계수의 비트량을 크게 감소시켰다.

그러나 MPEG-4 Part 10 표준에 있어서 움직임 벡터에 대한 압축률은 실질적으로 개선되지 못하였다. 왜냐하면 MPEG-4 Part 10 표준에서의 움직임 벡터 부호화 기법은 기존 압축 표준들에서 사용되고 있는 기법과 근본적으로 동일하기 때문이다. 그런데, 오늘날 MPEG-4 Part 10 표준에서 움직임 벡터 부호화에 소요되는 비트량은 전체 비트량에 대하여 무시할 수 없는 비중을 차지하고 있다. 그림 1은 Foreman 동영상에 있어서 주요 구문 요소들에 대한 비트량 분포를 보여준다.

그림 1로부터 저 비트율의 압축 환경(Qp가 큰 영역)에서 움직임 벡터의 비트량이 전체 비트량에 대하여 큰 비중을 차지한다는 것을 알 수 있다. 따

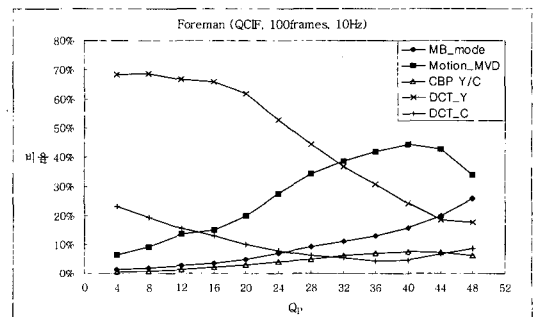


그림 1. MPEG-4 Part 10 표준의 주요 구문 요소들에 대한 비트량 분포

라서 MPEG-4 Part 10 표준의 성능 개선을 위해서는 효과적인 움직임 벡터 부호화 방식 개발이 필요하다.

MPEG-4 Part 10 동영상 압축 표준에서는 이미 압축된 주위 블록들이 지니고 있는 움직임 벡터들의 중간값(Median)을 현재 블록의 움직임 벡터에 대한 추정치로 이용한다. 그리고 이때 얻어진 MVD를 Exp-Golomb 부호어를 이용하여 부호화한다. 그림 2는 MPEG-4 Part 10 표준에서의 움직임 벡터 부호화 방식을 나타낸다. 여기서, i 는 가로와 세로 방향을 나타내는 첨자이다. 그림 2에서 MVC_i 는 압축할 현재 블록에 대한 i 방향 움직임 벡터를 나타낸다. 그리고 MVP_i 와 MVD_i 는 주위 블록들의 움직임 벡터들로부터 얻어지는 추정 벡터와 실제로 압축되는 MVD를 각각 나타낸다. 이것은 식(1)과 식(2)와 같이 표현된다. MPEG-4 Part 10 표준에서는 이렇게 얻어진 MVD를 가로, 세로 방향 각각에 대하여 항상 독립적으로 압축한다.

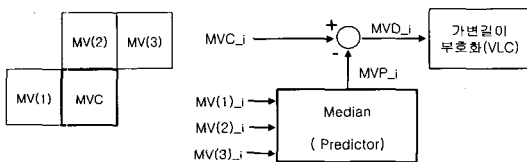


그림 2. MPEG-4 Part 10 표준에서의 움직임 벡터 부호화 방식.

$$MVP_i = \text{med}\{MV(1)_i, MV(2)_i, MV(3)_i\} \quad (1)$$

for i = x, y

$$MVD_i = MVC_i - MVP_i \quad (2)$$

III. 기존의 움직임 벡터 부호화 방식

3.1 최소 비트율 방식⁽⁶⁾

일반적으로 앞에서 설명한 표준 부호화 방식은 큰 움직임을 지니고 있는 동영상에 대하여 우수한 압축 성능을 보여주지 못하는 문제점을 지니고 있다. 압축할 동영상이 큰 움직임을 지니고 있을 경우, 중간값 예측에 의하여 얻어지는 MVD는 대체적으로 큰 값을 지니게 되어 효과적인 압축을 수행하지 못한다. 이것은 추정 벡터의 선정에 있어서 중간값 예측이 동영상의 특성을 반영하지 못하기 때문이다.

최소 비트율 방식은 주어진 동영상의 움직임 정

도와 복잡도에 따라 움직임 벡터를 효과적으로 추정함으로써 부호화 효율의 향상을 시도하였다. 이 방식에서는 먼저 주위 블록들의 움직임 벡터들에 있어서 최대, 최소 차이를 구하여 현재 블록의 움직임 정도를 예측한다. 식 (3)은 이 과정을 나타낸다.

$$\Delta_i = MV_{\max_j} - MV_{\min_j}$$

$$\text{where } MV_{\max_j} = \max\{MV(1)_i, MV(2)_i, MV(3)_i\}$$

$$MV_{\min_j} = \min\{MV(1)_i, MV(2)_i, MV(3)_i\} \quad (3)$$

그리고 Δ_i 가 주어진 임계값보다 작다면 현재 블록의 움직임이 작다고 간주하고 기존 표준 방식에 따라 부호화를 수행한다.

그러나, Δ_i 가 임계값보다 큰 경우에는 MVD를 최소로 하는 주위 블록의 움직임 벡터를 추정 벡터로 설정하고 이때 얻어진 최소 MVD를 부호화한다. 식 (4)와 식(5)는 이와 같은 과정을 나타낸다.

$$k = \arg \{ \min_j (|MVC_i - MV(j)_i|) \}$$

for j = 1, 2, 3 (4)

$$MVD_i = MVC_i - MVP_i = MVC_i - MV(k)_i \quad (5)$$

식(4)와 식(5)는 기존 표준 방식에 비하여 작은 MVD가 가변길이 부호화됨을 보여준다. 그런데 이 방식에서는 3가지 움직임 후보 벡터들 가운데 어느 것이 추정 벡터로 이용되었는지에 대한 모드 정보가 복원단으로 전송되어야 한다. 따라서, 각 방향별로 부가적인 모드 비트가 요구된다. 그러나 이러한 비트 증가는 MVD 크기 감소로 인한 비트 감소량으로 충분히 상쇄되어진다. 따라서 이 방식은 움직임이 많고 복잡한 영상에서 우수한 성능을 발휘한다. 그러나 최소 비트율 방식은 움직임이 작은 경우에 기존 표준 방식이 적용되기 때문에 효과적인 압축이 수행되지 못하는 단점을 지니고 있다.

3.2 2차원 결합 부호화 방식⁽⁷⁾

최근에 화상회의(Video Conference)나 비디오폰(Video Phone) 서비스와 같은 응용 분야가 많은 주목을 받고 있다. 이들 분야에서 다루어지는 동영상들은 특성상 움직임이 비교적 작고 단순한 형태를 지닌다. 그러나 표준 부호화 방식과 최소 비트율 방식은 이 같은 특징을 고려하지 않고 압축을 수행하기 때문에 효과적인 압축이 이루어지지 못한다.

일반적으로 움직임이 작은 동영상에 있어서 MVD는 예측 기법에 관계없이 작은 값을 지니며 대부분 (0,0) 근처 영역에 분포한다. 따라서 가로와 세로 방향의 MVD들을 결합하여 2차원으로 부호화 할 경우에 성능 향상을 기대할 수 있다⁵⁾.

이 사실에 기초하여 Kimata는 MPEG-4 Part 10 표준에 있어서 가로와 세로 방향의 MVD를 결합한 후 이를 가변길이 부호화하는 방식을 제안하였다. 그림 3은 (MVDx, MVDy)에 대한 Exp-Golomb 기반의 결합 부호어를 나타낸 것이다.

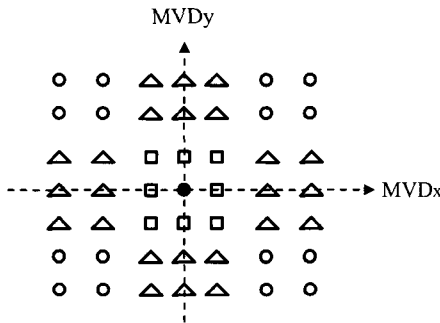
그림 3에서 p는 가로와 세로 방향을 의미하는 1bit의 플래그이고 b의 열은 Exp-Golomb 부호어와 같은 정보 비트부를 나타낸다. 그리고 표 1은 그림 3의 결합 부호어를 적용하여 얻어진 2차원 움직임 벡터 부호화 성능을 보여준다. 표 1에서 양의 값은 기존 표준 부호화 방식에 대한 상대적인 비트 이득을 나타낸다. 표 1로부터 Hall, Grandma, Container 동영상에서 2차원 결합 부호화가 효과적임을 알 수 있다.

그러나 2차원 결합 부호화 방식이 기존 표준 부호화 방식에 비하여 항상 우수한 것은 아니다. 표 1

표 1. 결합 부호어를 이용한 2차원 움직임 벡터 부호화 성능(%)

해상도	Sequence	Qp=28	Qp=32
QCIF (30Hz)	Hall	3.74	2.84
	Container	5.08	3.30
	Grandma	0.35	0.10
	Foreman	-2.42	-2.35
	Claire	-1.31	-1.20
	News	-2.39	-2.67

에서 Foreman, News의 동영상에서 비트 손실이 발생함을 알 수 있다. 이것은 중간값 예측에 의하여 구해지는 MVD가 영상에 따라 그 분포 특성이 달라지기 때문이다. MVD가 충분히 작지 않을 경우에 그림 3의 결합 부호어를 적용하는 것은 큰 효과를 발휘하지 못한다. 더구나 기존 독립 부호어와 비교할 때 결합 부호어는 (MVDx, MVDy)의 위치에 따라 비트 이득과 손실이 결정된다. 따라서 2차원 결합 부호화를 이용하여 압축 효율을 증가시키려면 가능한 한 MVD가 (0,0)에 분포되어야 한다. 그리고 비트 손실이 발생하는 경우에는 기존의 독립 부호화 방식을 적용하여야 한다.



분류	(MVDx, MVDy)	기존 독립 부호어	2차원 결합 부호어	이득
C1 ●	(0,0)	(1,1)	(1)	1
C2 ◻	(-1/4, -1/4)	(010,010)	(01000)	1
	(-1/4, 0)	(010,1)	(01001)	-1
	(-1/4, 1/4)	(010,011)	(01010)	1
	(0, -1/4)	(1,010)	(01011)	-1
	(0, 1/4)	(1,011)	(01100)	-1
	(1/4, -1/4)	(011,010)	(01101)	1
	(1/4, 0)	(011,1)	(01110)	-1
	(1/4, 1/4)	(011,011)	(01111)	1
C3 ▲	(x , 0)	(001bb,1)	(001bb,1,p)	-1
	(0, y)	(1,001bb)	(001bb,1,p)	-1
	(x , 1/4 y)	(001bb,01b)	(001bb,01b,p)	-1
	(1/4 x , y)	(01b,001bb)	(001bb,01b,p)	-1
C4 ○	Others	(001bb,001bb)	(001bb,001bb)	0

그림 3. (MVDx, MVDy)에 대한 Exp-Golomb 기반의 결합 부호어.

IV. 제안하는 움직임 벡터 부호화 방식

앞에서 살펴 본 사실을 토대로 효과적인 움직임 벡터 부호화를 수행하기 위해서는 다음의 사항들이 고려되어야 함을 알 수 있다.

- 1) MVD를 작게 하기 위한 추정 방식.
- 2) 결합 부호화와 독립 부호화의 선별적 적용을 위한 기준.
- 3) 부가적인 메모리 및 계산량의 증가없는 압축, 복원.

이 중에서 1)은 최소 비트율 예측이 해당이 될 수 있다. 왜냐하면 최소 비트율 예측은 영상의 움직임 특성에 관계없이 최소 MVD를 제공하기 때문이다. 그러나 2)에 대하여는 지금까지 제시된 해결책이 존재하지 않는다. 본 논문에서는 MPEG-4 Part 10 표준의 특성과 움직임 벡터의 분포를 분석함으로써 2)의 기준을 제시한다.

4.1 매크로블록 유형에 따른 움직임 벡터 분포 특성

이전 동영상 표준안과는 달리 MPEG-4 Part 10 표준에서는 매크로블록 유형(MB_type)이라는 새로

운 구분 요소가 채택되었다. 매크로블록 유형은 움직임 추정과 보상과정에 있어서 매크로블록의 크기 및 형태를 나타낸다. 일반적으로 프레임 내에서 움직임이 없고 평탄한 부분에서는 16x16 매크로블록 유형이 빈번하게 선택되는 반면 움직임이 많거나 물체와 배경의 경계가 되는 에지 부분에서는 다른 유형들이 선택된다. 따라서 16x16 매크로블록 유형이 선택될 경우 MVD가 (0,0)부근에 존재할 가능성이 매우 크다. 이것은 MVD의 실제 분포를 조사해봄으로써 확인할 수 있다.

표 2는 16x16 매크로블록 유형과 16x8 매크로블록 유형에서의 MVD분포를 보여준다. 표 2의 (a)에서 (0,0)의 발생 빈도가 50% 이상임을 알 수 있다. 이것은 기존 독립 부호화보다 2차원 결합 부호화가 효과적임을 의미한다. 반면에 (b)에서는 (0,0)의 발생 빈도가 50% 이하이며 그림 3의 결합 부호어에서 비트 손실이 발생하는 위치에서의 발생 빈도들의 합 또한 적지 않다.

따라서 이 경우 2차원 결합 부호화보다는 기존의 독립 부호화를 적용하는 것이 타당할 것이다.

본 논문에서는 이와 같은 사실에 기초하여 16x16 매크로블록 유형에 따라 결합 부호화와 독립 부호화의 적용 여부를 결정한다. 즉, 압축할 현재 블록이 16x16 매크로블록 유형일 경우 결합 부호화를 선택한다. MPEG-4 Part 10 표준에서 매크로블록 유형 구분요소가 움직임 벡터 구분요소보다 앞서서 압축, 전송되므로 복호화기에서 문제가 야기되지 않는다.

표 2. 매크로블록 유형에 따른 MVD 분포(News, Qp32, 10Hz/30Hz)

$ MVDy \backslash MVDx $	0	1/4	2/4	3/4
0	62.6/70.6	5.4/7.3	2.5/2.4	2.2/1.4
1/4	9.8/10.4	4.3/2.7	0.2/0.3	0.3/0.2
2/4	1.9/0.9	0.9/0.2	0.3/0.2	0.0/0.0
3/4	0.9/0.4	0.4/0.1	0.2/0.1	0.0/0.0

(a) 16x16

$ MVDy \backslash MVDx $	0	1/4	2/4	3/4
0	26.6/28.7	13.2/16	2.7/4.0	4.3/3.8
1/4	12.3/15.2	5.8/7.4	2.1/1.2	1.6/0.9
2/4	6.7/5.6	1.9/1.6	0.5/0.4	0.7/0.4
3/4	2.4/1.8	1.6/1.1	0.5/0.3	0.5/0.3

(b) 16x8

4.2 Δ_i 에 따른 움직임 벡터 분포 특성

최소 비트율 방식에서는 식(3)에서 정의한 Δ_i 를 이용하여 현재 블록의 움직임 정도를 예측하였다. Δ_i 값이 크다면 현재 블록의 움직임이 크다고, 그와 반대로 값이 작으면 움직임이 작다고 간주하였다. 그러나 최소 비트율 방식은 기존 독립부호화를 기반으로 하기 때문에 2차원 결합 부호어가 효과를 발휘하는 Δ_i 의 영역을 제시하지 못하고 있다.

결합 부호화가 효과적인 Δ_i 의 범위를 결정하기 위하여 MVD 분포를 조사함으로써 표 3과 같은 결과를 얻었다. 표 3에서 MVD가 (0,0)에서 집중적으로 발생하고 이득이 발생하는 위치의 발생 빈도 총합이 50% 이상임을 알 수 있다. 이것은 2차원 결합 부호화를 적용한다면 압축 효율을 보다 더 향상시킬 수 있음을 암시한다. 따라서 가로와 세로 방향 모두 $\Delta_i \leq 0.5$ 일 때 결합 부호화를 적용하는 것이 효과적이다.

이상의 앞에서 살펴본 바와 같이 2차원 결합 부호화는 현재 매크로 블록의 유형이 16x16 이거나

표 3. $\Delta_i \leq 0.5$ 에서의 MVD 분포(News, Qp32, 10Hz/30Hz)

$ MVDy \backslash MVDx $	0	1/4	2/4	3/4
0	56.9/53.6	9.6/10.3	2.8/2.8	1.1/1.8
1/4	12.6/10.4	3.1/3.2	0.8/1.1	0.2/0.4
2/4	2.5/2.5	0.4/1.5	0.14/0.4	0.0/0.2
3/4	0.8/0.9	0.7/0.5	0.0/0.3	0.1/0.1

표 4. 2차원 결합부호화에서 사용된 부호어

분류	(MVDx, MVDy)	2차원 결합 부호어
C1 ●	(0,0)	(1)
C2 □	(-1/4, -1/4)	(0111111)
	(-1/4, 0)	(0110)
	(-1/4, 1/4)	(0111110)
	(0, -1/4)	(01110)
	(0, 1/4)	(0101)
	(1/4, -1/4)	(0111101)
	(1/4, 0)	(0100)
	(1/4, 1/4)	(0111100)
C3 ▲	(x , 0)	(001bb, 1, p)
	(0, y)	(001bb, 1, p)
	(x , 1/4)	(001bb, 01b, p)
	(1/4 , y)	(001bb, 01b, p)
C4 ○	Others	(001bb, 001bb)

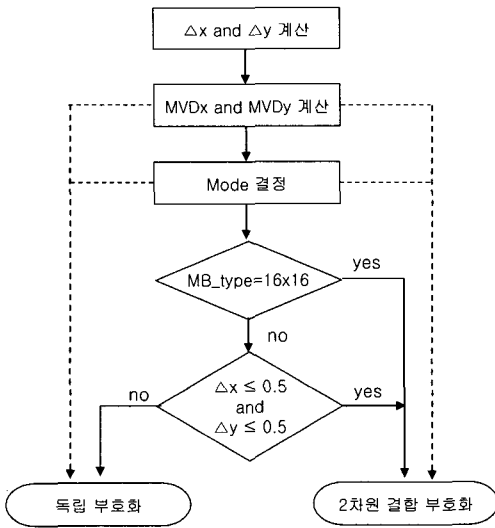


그림 4. 제안하는 부호화 알고리즘의 순서도

Δ_i 가 0.5이하인 경우에 효과적이다. 본 논문에서는 이러한 사실을 토대로 효과적인 움직임 벡터 부호화 알고리즘을 제안한다. 제안 알고리즘에서는 최소 비트율 예측에 의하여 MVD가 계산되며 독립 부호화와 결합 부호화의 적용 여부는 매크로블럭 유형과 Δ_i 값에 따라 결정된다.

그림 4는 제안하는 움직임 벡터 부호화 알고리즘의 순서도이다. 그림 4에서 독립 부호화는 각 방향의 MVD를 Exp-Golomb 부호어를 이용하여 독립적으로 압축한다. 그리고 2차원 결합 부호화는 표 4에 제시된 결합 부호어를 이용하여 압축을 수행한다. 표 4의 결합 부호어는 메모리를 사용하지 않고 압축과 복원에 이용될 수 있다.

V. 실험 결과 및 고찰

기존 방식들과 제안 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. 이와 같은 이유로 본 논문에서는 제안 알고리즘의 우수성을 확인하기 위하여 JM8.2 부호화기^[8]를 통하여 기존 방식과 제안 방식간의 압축 성능을 비교, 평가하였다. 모의 실험은 300프레임으로 구성된 다양한 QCIF 및 CIF형식의 동영상 시퀀스들에 대하여 수행되었다. 이때 탐색 범위는 -16.0~15.5로 설정하였다.

표 5는 움직임 벡터에 대한 기존 최소 비트율 방

식과 제안 방식간의 성능 비교를 보여준다. 표 5의 값은 기존 표준 방식에서 얻어진 비트량에 대한 상대적인 비트 이득을 의미한다. 표 5는 제안 방식이 기존 표준 방식에 비해 항상 비트 이득을 가져옴을 보여준다. 그리고 최소 비트율 방식에 비해 약 3% 정도 우수한 성능을 지니고 있음을 알 수 있다. 특히, 움직임이 작은 동영상에서는 평균보다 높은 이득이 발생함을 알 수 있다. 또한 표 1의 기존 결합 부호화 방식의 성능과 비교해 볼 때 제안 방식은 다양한 동영상들에 대하여 항상 우수하며 평균 5% 이상의 압축 이득을 보여준다. 이것은 결합 부호화의 적용 기준이 효과적으로 작용하였음을 의미한다.

표 5. 움직임 벡터에 대한 성능 비교 (%)

해상도	Sequence	10Hz		30Hz	
		[6]	제안 방식	[6]	제안 방식
QCIF	Claire	2.13	4.74	0.56	5.65
	Container	-0.16	5.17	0.40	10.30
	Coastguard	4.04	5.50	3.82	6.47
	Foreman	3.37	4.34	1.89	3.73
	Grandma	1.92	5.71	0.25	6.56
	Hall	4.83	8.55	1.65	9.26
	News	5.70	7.38	4.33	7.13
	Silent	3.90	5.13	4.09	6.05
	Average	3.22	5.82	2.12	6.89
CIF	Akiyo	1.88	3.38	0.81	5.08
	Coastguard	4.39	5.29	2.20	4.97
	Foreman	3.51	3.97	2.48	4.34
	News	4.77	5.61	4.05	6.18
	Average	3.64	4.56	2.38	5.14

(a) Qp=28

해상도	Sequence	10Hz		30Hz	
		[6]	제안 방식	[6]	제안 방식
QCIF	Claire	1.69	3.30	0.61	4.81
	Container	0.08	5.87	0.22	10.39
	Coastguard	3.47	4.86	3.37	5.78
	Foreman	2.08	2.58	1.22	2.36
	Grandma	1.84	4.40	0.27	5.47
	Hall	3.56	6.18	1.42	7.08
	News	3.94	5.28	3.32	5.51
	Silent	2.85	3.86	3.14	4.81
	Average	2.44	4.54	1.70	5.78
CIF	Akiyo	1.50	3.37	1.00	4.50
	Coastguard	2.69	3.80	1.71	4.18
	Foreman	2.49	3.33	2.02	3.44
	News	3.12	4.33	2.85	4.50
	Average	2.45	3.71	1.90	4.16

(b) Qp=32

한편, 표 6은 전체 압축 비트량에 대한 성능 비교를 보여준다. 표 6은 전체 비트량 측면에서 제안 방식이 최소 비트량 방식에 비하여 약 0.5%이상의 비트 이득을 가져옴을 보여준다.

표 6. 전체 압축 비트량에 대한 성능 비교(%)

해상도	Sequence	10Hz		30Hz	
		[6]	제안 방식	[6]	제안 방식
QCIF	Claire	0.53	1.20	0.13	1.18
	Container	-0.05	0.69	0.03	1.10
	Coastguard	0.80	1.09	0.61	1.03
	Foreman	0.98	1.26	0.57	1.11
	Grandma	0.43	1.31	0.04	1.23
	Hall	1.03	1.81	0.28	1.62
	News	1.43	1.85	1.00	1.65
	Silent	1.06	1.41	1.17	1.73
	Average	0.78	1.33	0.48	1.33
CIF	Akiyo	0.48	1.12	0.19	1.14
	Coastguard	1.1	1.33	0.36	0.82
	Foreman	0.68	0.93	0.64	1.13
	News	1.15	1.52	1.06	1.61
	Average	0.85	1.23	0.56	1.18

(a) Qp=28

해상도	Sequence	10Hz		30Hz	
		[6]	제안 방식	[6]	제안 방식
QCIF	Claire	0.47	0.91	0.17	1.02
	Container	0.02	0.98	0.02	1.34
	Coastguard	0.77	1.07	0.71	1.22
	Foreman	0.67	0.84	0.41	0.80
	Grandma	0.47	1.09	0.06	1.09
	Hall	0.85	1.48	0.27	1.37
	News	1.06	1.41	0.90	1.46
	Silent	0.80	1.05	0.93	1.40
	Average	0.64	1.10	0.43	1.21
CIF	Akiyo	0.43	1.00	0.27	1.19
	Coastguard	0.64	0.91	0.32	0.79
	Foreman	0.55	0.74	0.63	1.09
	News	0.79	1.09	0.83	1.30
	Average	0.60	0.94	0.51	1.09

(b) Qp=32

VI. 결론

본 논문에서는 MPEG-4 Part 10 표준에서의 움직임 벡터의 부호화 효율을 개선 위한 새로운 알고리즘을 제안하였다. 기존의 부호화 방식들에서는 동

영상의 움직임 특성이 충분히 반영되지 못하여 효과적인 압축을 수행하지 못하였다. 이를 해결하기 위하여 본 논문에서는 MVD의 분포를 분석하여 움직임 특성을 예측할 수 있는 조건을 제시하였다. 또한 MPEG-4 Part 10의 구분 요소들 중에서 매크로 블록 유형이 움직임 특성에 관한 정보를 지니고 있음을 보였다. 이를 토대로 제안 알고리즘에서는 독립 부호화 방식과 결합 부호화 방식을 선택적으로 적용함으로써 움직임 벡터를 압축한다. 모의 실험 결과는 영상의 움직임 특성에 관계없이 움직임 벡터 및 전체 비트량에 대하여 평균 3%와 0.5%이상의 압축률 향상이 제안 방식에서 이루어짐을 보여준다. 또한 제안 방식은 부가적인 메모리 및 과도한 계산량을 필요로 하지 않은 잇점을 지닌다.

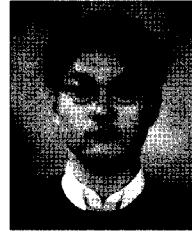
참고 문헌

- [1] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 ISO/IEC 14496-10 AVC)," Doc. JVT- G050.doc, Mar. 2003.
- [2] J. Ribas-Corbera and D. L. Neuhoff, "Optimizing motion-vector accuracy in block-based video coding," IEEE Trans. on Circuits and Systems for Video Technology, vol. 11, no. 4, pp. 497-511, Apr. 2001.
- [3] J. Saste, A. Ferreras, and J. F. Hernandez-Gil, "Motion vector size- compensation based method for very low bit-rate video coding," IEEE Trans. on Circuits and Systems for Video Technology, vol. 10, no.7, pp. 1192-1197, Oct. 2000.
- [4] G. Cote, M. Gallant, and F. Kossentia, "Semi-fixed-length motion vector coding for H.263-based low bit rate video compression," IEEE Trans. on Image Processing, vol. 8, no. 10, pp. 1451-1455, Oct. 1999.
- [5] G. Y. Yu and C.T. Chen, "Two-Dimensional Motion Vector Coding for Low Bitrate Videophone Applications", Proceedings of IEEE International Conference on Image Processing, pp. 414-417, Oct. 1995.

- [6] S. D. Kim and J. B. Ra, "An Efficient Motion Vector Coding Scheme Based on Minimum Bitrate Prediction", IEEE Trans. on Image Processing, vol. 8, no. 8, pp.1117-1119, Aug. 1999.
- [7] H. Kimata, M. Kitahara, Y. Yashima, "3D motion vector coding with block base adaptive interpolation filter on H.264," in Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP '03), Vol. 3, pp. 633-636, Apr. 2000.
- [8] Reference S/W, [http://iphome.hhi.de /suehring/tml/download/old_jm/jm82.zip](http://iphome.hhi.de/~suehring/tml/download/old_jm/jm82.zip), May, 2005.

문 용 호 (Yong-ho Moon)

정회원



1992년 2월 부산대학교 전자공학과(공학사)

1994년 2월 부산대학교 일반대학원 전자공학과(공학석사)

1998년 8월 부산대학교 일반대학원 전자공학과(공학박사)

1998년 9월~2001년 8월 삼성전

자 중앙 연구소 Digital Media Lab. 책임연구원

2001년 9월~2003년 2월 부산대학교 산업자동화 및 정보통신분야 인력양성 사업단(BK21) (기금교수:조교수대우)

2003년 3월~현재 부산외국어대학교 디지털미디어학부(조교수)

<관심분야> 동영상 압축, 영상처리, 디지털시스템 설계 및 VLSI