

일반논문-11-16-1-13

## 엔트로피 코딩 기반의 분산 비디오 코딩을 위한 블록 기반 복잡도 분배

유성은<sup>a)</sup>, 민경연<sup>a)</sup>, 심동규<sup>a)‡</sup>

### Complexity Balancing for Distributed Video Coding Based on Entropy Coding

Sung-Eun Yoo<sup>a)</sup>, Kyung-Yeon Min<sup>a)</sup>, and Dong-Gyu Sim<sup>a)‡</sup>

#### 요 약

본 논문은 엔트로피 코딩 기반 분산 비디오 코딩 시스템에서의 복잡도 분배 기법을 제안한다. 제안하는 방법은 복호화기의 복잡도 감소를 위하여 채널 코더 대신 엔트로피 코더를 이용하며, 저 복잡도로 높은 부호화 효율을 얻기 위한 블록 단위 복잡도 분배 방법을 수행한다. 제안하는 분산 비디오 복호화기는 움직임 추정을 수행하여 측정된 움직임 벡터를 부호화기로 전송하고, 부호화기에서는 복호화기로부터 수신된 움직임 벡터를 보정하여 보다 정확한 움직임 추정을 수행한다. 움직임 벡터의 보정을 수행 시, 수신된 움직임 벡터와 예측 움직임 벡터를 이용하여 최적의 예측 움직임 벡터를 결정하며, 움직임 벡터와 예측 움직임 벡터의 차에 따라 범위를 조절함으로써 블록의 복잡도를 적응적으로 할당한다. 제안하는 부호화기는 H.264/AVC의 부호화기의 복잡도에 비교하여 11.8% 감소하였고, 제안하는 복호화기는 기존의 분산 비디오 시스템의 복호화기 복잡도보다 99%감소되었다.

#### Abstract

In this paper, a complexity-balancing algorithm is proposed for distributed video coding based on entropy coding. In order to reduce complexity of DVC-based decoders, the proposed method employs an entropy coder instead of channel coders and the complexity-balancing method is designed to improve RD performance with minimal computational complexity. The proposed method performs motion estimation in the decoder side and transmits the estimated motion vectors to the encoder. The proposed encoder can perform more accurate refinement using the transmitted motion vectors from the decoder. During the motion refinement, the optimal predicted motion vectors are decided by the received motion vector and the predicted motion vectors and complexity load of block is allocated by adjusting the search range based on the difference between the received motion vector and the predicted motion vectors. The computational complexity of the proposed encoder is decreased 11.9% compared to the H.264/AVC encoder and that of the proposed decoder are reduced 99% compared to the conventional DVC decoder.

Keyword : Distributed video coding, Motion vector refinement, Complexity balancing, Predictive motion vector, Bitrate control

a) 광운대학교 컴퓨터공학과

Kwangwoon University

‡ 교신저자 : 심동규 (dgsim@kw.ac.kr)

※ 본 연구는 지식경제부, 방송통신위원회 및 한국산업기술평가관리원의 산업원천기술개발사업(정보통신)의 일환으로 수행하였음. [KI002142, 차세대 모바일 영상서비스를 위한 초경량 비디오 부호화 원천기술개발]

· 접수일(2010년11월23일), 수정일(2011년1월11일), 게재확정일(1월11일)

## I. 서론

최근 프로세서, 저장장치 및 전송기술이 급격하게 발달함에 따라 멀티미디어 처리 기능을 가진 휴대 기기의 생산 및 보급이 활발하게 이루어졌다<sup>[1-4]</sup>. 사용자는 휴대기기를 이용하여 멀티미디어 데이터를 취득하고 콘텐츠를 제작 및 전송, 보급할 수 있게 된 것이다. 이제는 사용자가 단순히 생산된 멀티미디어 데이터를 소비하는 것이 아니라 사용자 스스로 멀티미디어 데이터를 생산하고 배포하는 시대가 된 것이다. 하지만, 기존의 표준 부호화 방법인 ISO MPEG-x나 ITU-T H.26x 등의 표준 비디오 부/복호화 방법은 저장장치 혹은 스트리밍 방송 서비스를 위해 개발된 기술로, 하나의 생산주체가 다수의 소비자에게 데이터를 전송하는 방법에 최적화 되어 발전되었다. 따라서 서비스 제공할 때 비용 절감을 위하여 생산주체는 높은 부호화 효율로 멀티미디어 데이터를 처리하여 이를 서비스하기 때문에, 부호화기의 성능은 복잡도가 아닌 부호화 효율에 그 초점이 맞춰 발전되어 왔다<sup>[5-6]</sup>. 이에, 부호화기의 복잡도는 낮고 부호화기의 복잡도가 높아, 복호화를 수행하는 동안에는 배터리의 소비가 적고 복호화 시간이 단축되지만, 부호화를 수행하는 동안에는 배터리의 소비가 많고 부호화 시간이 길어진다. 이러한 부호화 방법은 개개인의 콘텐츠 제작에서 이용되는 무선 센서 네트워크 장치에는 적절한 방법이 되기 어렵다<sup>[7]</sup>. 따라서 휴대기기의 제한된 성능 안에서 다양한 멀티미디어 데이터를 처리하기 위해서 짧은 부호화 시간과 낮은 전원소비가 가능한 부호화기의 필요성이 대두되고 있다. 이에 대한 연구의 일환으로 새로운 구조의 코덱인 분산 비디오 코딩 방법이 연구되고 있다<sup>[1-4],[7-9]</sup>. 분산 비디오 부호화는 부호화기의 복잡도를 부호화기로 분산시켜 부호화기의 복잡도를 낮추는 방법으로서, 배터리의 소비가 적고 부호화 시간이 짧아 이동식 비디오 장치에 적합하다. 그러나 분산 비디오 부호화 기술의 낮은 압축 성능, 높은 부호화기의 복잡도와 피드백 채널의 존재 등 해결해야 할 문제점들이 아직은 존재한다.

본 논문에서는 H.264/AVC와 같이 높은 부호화기 복잡도를 갖는 비디오 부호화 방법과 낮은 부호화 성능과 높은 복호화 복잡도를 갖는 분산 비디오 코딩 방법의 문제점들을 해결하기 위하여, 블록 기반으로 복잡도 분배를 수행하

는 엔트로피 기반의 분산 비디오 코딩 방법을 제안한다. 복호화기의 복잡도를 감소시키기 위하여 제안한 방법은 기존의 분산 비디오 코딩 방법에서 사용한 채널 코딩을 대신하여 엔트로피 코딩을 사용한다. 또한, H.264/AVC의 높은 부호화기 복잡도를 감소시키기 위하여 복호화기에서 움직임 추정을 하고, 측정된 움직임 벡터를 부호화기로 전송을 수행한다. 수신된 움직임 벡터와 주변 블록들의 움직임 벡터의 중간값으로 결정된 예측 움직임 벡터 중 최적의 예측 움직임 벡터를 결정한다. 또한, 수신된 움직임 벡터와 예측 움직임 벡터의 차에 따라 복잡도를 할당함으로써, 할당된 복잡도 내에서 부호화 효율을 최적화 한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 연구에 대하여 살펴보고, 3장에서 제안하는 알고리즘에 대하여 소개한다. 4장에서는 제안하는 알고리즘에 대한 실험방법과 실험결과에 대해서 소개하고 분석한다. 마지막으로 5장에서는 결론을 맺겠다.

## II. 기존 분산 비디오 코딩 및 H.264/AVC 압축 방법

### 1. 기존 분산 비디오 코딩

휴대기기에 적합한 비디오 압축을 위하여 분산 비디오 코딩 기술이 연구되고 있다. Slepian-Wolf와 Wyner-Ziv의 정보이론 관점으로 부터 발전되어온 분산 비디오 코딩은 UC Berkeley 대학의 K. Ramchandran 교수 연구팀에서 개발된 PRISM(Power efficient, Robust, hIgh compression, Syndrome based Multimedia coding)방법과 미국 스탠포드 대학의 B. Girod 교수 연구팀에 의해 개발된 Wyner-Ziv 기반 분산 비디오 부호화 방법이 있다. 제안한 방법은 Wyner-Ziv 부호화 방법에 근간을 두고 있으며, Wyner-Ziv 부호화 방법의 전체 구조도는 그림 1과 같다<sup>[1]</sup>.

Wyner-Ziv 부호화 구조는 그림 1과 같이 피드백 채널을 가지는 특징을 가지며, 프레임을 두 종류로 구분하여 서로 다른 방법으로 부호화를 수행한다. Xt에 해당하는 프레임은 채널 코더를 이용해 부호화 되며, 이 프레임을 WZ 프레

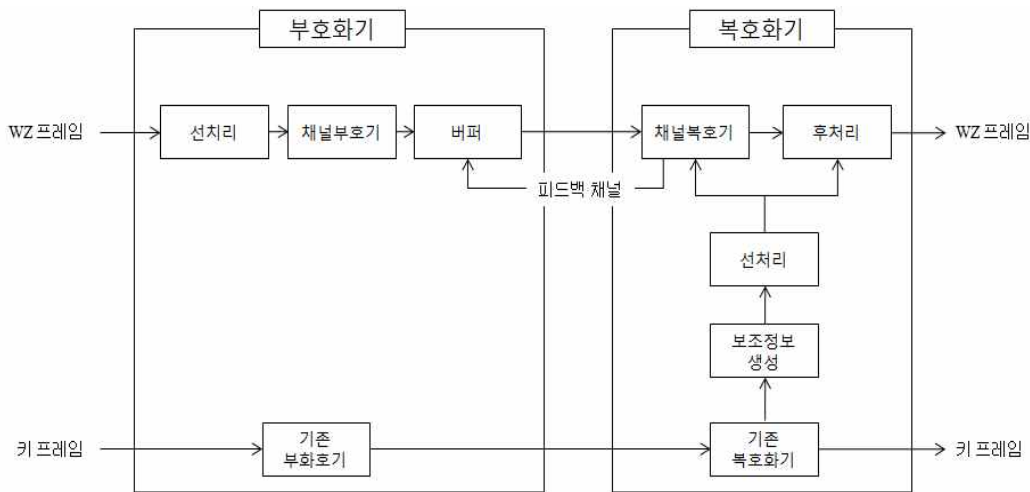


그림 1 . 기존의 분산 비디오 코딩 방법  
Fig. 1. Block diagram of conventional distributed video coding

임이라 한다. 또한,  $X_{t-1}$ 과  $X_{t+1}$  프레임은 H.264/AVC 등 기존 부호화 방법의 화면 내 예측 방법을 통해 부호화되며, 이 프레임을 키 프레임이라 한다. 키 프레임이 복호화기로 전송되면, 복호화기는 키 프레임을 기존 H.264/AVC 등의 복호화 방법을 통하여 복원한다. 이렇게 복원된 두 장의 키 프레임을 기반으로 움직임 추정 보간법을 수행하여 현재 프레임  $X_t$ 에 상응하는 보조정보를 만들어 낸다. 원본 프레임  $X_t$ 와 보조정보의 오차를 채널을 통해 발생되는 오류로 가정하고, 채널 코더를 이용하여 오류를 수정함으로써  $X_t$  프레임을 복원하게 된다. 채널 코딩은 일반적으로 성능이 높은 터보(Turbo) 코더와 low density parity check accumulation (LDPCA)가 사용된다. 채널 코더를 이용하여 현재 프레임의 오류를 수정할 때, 복호화기는 피드백 채널을 통하여 부호화기에 부분적으로 패리티를 요청하는데, 채널 복호화를 통해 얻어진 에러율이 충분히 작아질 때까지 부호화기에 패리티를 계속 추가로 요청하여 피드백 지연이 발생하게 된다<sup>11)</sup>.

분산 비디오 코딩에서 채널 코더를 이용한 복호화는 패리티 비트를 반복적으로 요청하여 복잡도가 높고, 복호화하고자 하는 프레임의 오류가 일정정도 높으면 부호화 효율이 나빠지는 단점이 존재한다.

## 2. H.264/AVC

H.264/AVC는 현재 가장 활발히 사용되고 있는 코덱 중의 하나로서 최신 비디오 압축 표준으로 ITU-T의 Video Coding Expert Group (VCEG)과 ISO/IEC의 Moving Picture Expert Group (MPEG)에서 공동으로 개발된 기술이다. H.264/AVC는 화면 내 예측, 화면 간 단방향 혹은 양방향 예측을 수행하여 높은 효율을 나타내며, 모든 가능한 예측을 수행해보고 비트율-왜곡 최적화 방법을 통하여 최적의 예측 방법을 선택한다.

또한 H.264/AVC는 다양한 예측 방법뿐만 아니라, 다중 참조 영상, 움직임 보상 과정에서의 높은 정밀도의 보간 필터, 다양한 블록 크기의 움직임 예측 모드, 인 루프(in-loop) 디블록킹 필터, 적응적인 엔트로피 기술 등의 다양한 기법을 사용함으로써 높은 부호화 효율을 가진다.

H.264/AVC는 소수의 생성자가 다수의 소비자에게 전송하는 상황에 최적화 되어 만들어진 압축 기술로서 부호화기의 복잡도가 복호화기의 복잡도에 비하여 상당히 높다. H.264/AVC는 프로파일을 정의하여 툴(tool)들을 여러 개의 집합으로 구성시켜 제공하지만 가장 복잡도가 낮은 Baseline 프로파일조차 휴대 기기의 저사양에 비교하여 높은 복잡도를 가지고 있다.

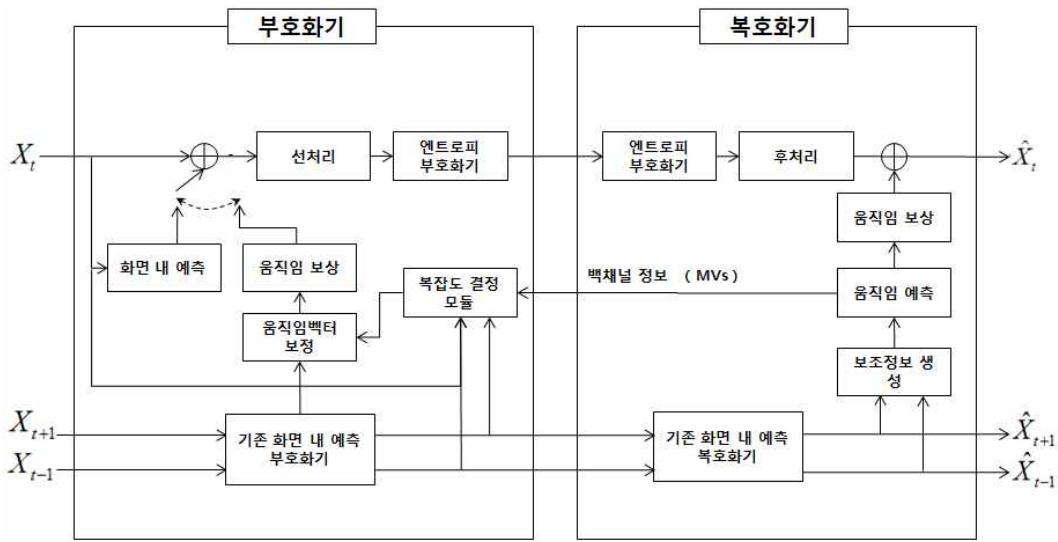


그림 2 . 제안하는 방법의 전체 블록도  
 Fig. 2. The block diagram of the proposed method

### III. 제안하는 방법

제안하는 방법은 기존의 분산 비디오 코딩의 단점인 복호화기의 높은 복잡도 문제를 해결하고 부호화기의 부호화 성능을 향상시키기 위하여 제안하였다. 복호화기의 높은 복잡도 문제를 해결하기 위해 기존의 분산 비디오 코딩에서 사용하는 채널 코더를 대신하여 엔트로피 코더를 사용하였다. 또한, 낮은 복잡도로 부호화 성능을 올리기 위한 블록 기반의 복잡도 분배 방법을 제안한다.

그림 2는 제안한 방법의 전체 구조도를 나타낸다. 그림 2에서의  $X_{t-1}$  프레임과  $X_{t+1}$  프레임이 H.264/AVC 화면 내 부호화 방법으로 부호화 되어 복호화기로 보내지면 복호화기는 두 프레임을 복호화하여 보조정보를 생성한다. 이 후 복호화기는 보조정보에서 키 프레임으로의 움직임 재추정을 통하여 움직임 벡터를 측정하고, 부호화기로 전송한다. 부호화기는 수신된 움직임 벡터를 보정하여 현재 프레임의 예측영상을 얻고 원 영상과의 차이값을 엔트로피 코딩하여 복호화기로 전송한다.

부호화기에서 움직임 벡터를 보정 시 예측 움직임 벡터의 신뢰도를 기반으로 적응적으로 복잡도를 할당한다. 예측 움직임 벡터의 신뢰도가 높은 블록에는 낮은 복잡도를

할당하고 신뢰도가 낮은 블록에는 높은 복잡도를 할당함으로써 낮은 복잡도로 높은 부호화 효율을 낼 수 있다.

기존의 분산 비디오 코딩 방법은 채널 코더를 이용하여 WZ(Wyner-Ziv)프레임을 부호화와 복호화를 수행한다. 채널 코더는 그 특성상 교차 확률이 임계치보다 높으면 복호화가 불가능하다. 이러한 이유로 오류가 없는 부분의 정보도 일정량 이상을 포함하여 채널 부호화와 복호화를 수행해야 하기 때문에 소요되는 비트의 효율성이 떨어진다. 또한, 보조정보의 교차 확률을 부호화기나 복호화기의 어느 쪽에서도 정확하게 측정할 수 없기 때문에 채널 코더의 성능이 감소되며 복잡도가 증가된다. 하지만 제안하는 엔트로피 코딩 기반의 분산 비디오 복호화기는 기존 분산 비디오 코딩에서 사용하는 채널 코더 대신에 엔트로피 코더를 사용함으로써 현재 프레임을 낮은 복잡도로 복원할 수 있고 부호화시 화면 내 부호화와 스킵모드의 사용으로 부호화 효율도 증가시킬 수 있다.

기존 H.264/AVC의 부호화기는 탐색 영역과 탐색 방법이 결정되면 모든 블록이 같은 탐색 영역을 가지고 움직임 추정을 수행한다. 따라서 움직임이 적어 좁은 탐색 영역으로도 최적의 블록을 탐색을 할 수 있는 경우와, 움직임이 많아 넓은 탐색 영역에서 움직임 추정을 수행해야하는 경

우를 고려치 않아 복잡도 측면에서 효율성이 낮다. 반면, 제안하는 방법은 예측 움직임 벡터의 신뢰도를 바탕으로 탐색영역을 조절하여 복잡도를 분배함으로써 낮은 복잡도로 효율있는 움직임 추정을 수행한다.

그림 3은 제안하는 블록 단위의 복잡도 분배 방법이다. 원본 블록과 보조정보 기반의 움직임 벡터 그리고 H.264/AVC기반의 예측 움직임 벡터를 입력하면 최적의 예측 움직임 벡터, 보정된 움직임 벡터와 스킵 가르침 비트가 결정된다. 여기서 복잡도란 블록에서 SAD (Sum of Absolute Difference)를 구하는 함수를 호출하는 횟수로 정의한다.

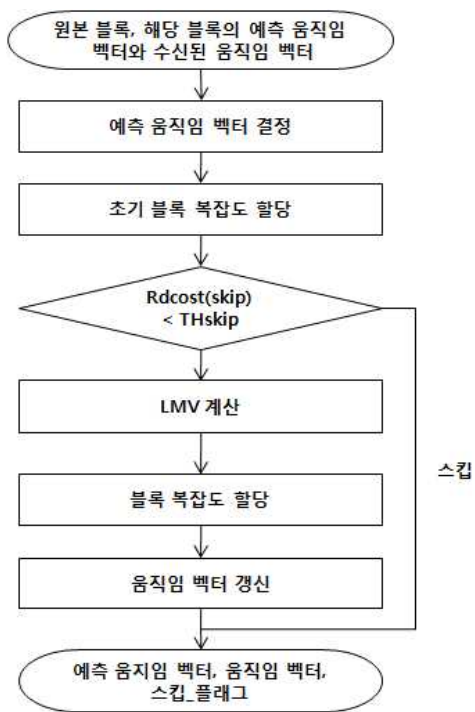


그림 3 제안하는 블록 단위의 복잡도 분배 방법  
Fig. 3 Complexity balancing of the proposed method

1. 최적의 예측 움직임 벡터 결정

수신된 보조정보 기반의 움직임 벡터와 주변 블록을 참조하여 결정된 예측 움직임 벡터 중 최적의 예측 움직임 벡터를 결정하고 이를 기반으로 탐색 위치를 결정한다. 그

림 4의  $PMV_{H.264/AVC}$ 는 H.264/AVC의 예측 움직임 벡터를 측정하는 방법으로 주변 블록의 움직임 벡터들의 중간값 필터를 이용해 결정된다. 주변 블록의 움직임 벡터는 이미 보정된 값이므로 현재 부호화 하는 블록의 움직임 벡터가 주변 블록과 유사하게 움직였을 경우 예측 움직임 벡터의 신뢰도가 높다<sup>[5]</sup>.

예측 움직임 벡터가 가리키는 블록과 원본 블록과의 SAD가 적을수록 예측 움직임 벡터의 주변에서 최적의 블록을 찾을 수 있는 확률이 높다. 따라서 다음의 식과 같이 H.264/AVC 기반의 예측 움직임 벡터와 수신된 움직임 벡터의 블록과 원 블록의 SAD를 비교하며 SAD가 더 작은 블록을 가리키는 움직임 벡터를 최종적으로 예측 움직임 벡터로 결정한다.

$$PMV = \begin{cases} MV_{SI} & \text{if } SAD_{MV_{SI}} \leq SAD_{PMV_{H.264/AVC}} \\ PMV_{H.264/AVC} & \text{otherwise} \end{cases} \quad (1)$$

식 (1)의  $SAD_{MV_{SI}}$ 는 수신된 움직임 벡터가 가리키는 블록과 원본 블록과의 SAD를 의미하며  $SAD_{PMV_{H.264/AVC}}$ 는 주변 블록을 참조하여 결정된 예측 움직임 벡터가 가리키는 블록과 원본 블록과의 SAD를 의미한다. 이렇게 결정된 예측 움직임 벡터는 복호화기에서 알 수 없기 때문에, 예측 움직임 벡터가 결정된 방법을 16x16 블록당 하나의 가르침 비트를 사용하여 복호화기에 전송한다. 복호화기에서는 이를 기반으로 예측 움직임 벡터를 생성하고 전송된 움직임 벡터 차분치를 이용하여 움직임 벡터를 복원할 수 있다.

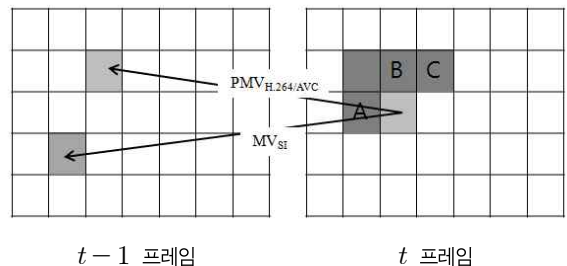


그림 4 . 제안한 방법의 예측 움직임 벡터 결정  
Fig. 4. PMV of the proposed method

### 2. 블록의 스킵모드 결정

예측 움직임 벡터가 결정 되면 이를 기반으로 블록 단위로 복잡도를 할당한다. 제안한 방법의 블록 단위 복잡도는 예측 움직임 벡터의 신뢰도에 따라서 결정되며, 수신된 움직임 벡터와 예측 움직임 벡터의 차가 적을수록 해당 블록의 움직임이 주변 블록의 움직임과 유사할 확률이 높기 때문에 높은 신뢰도를 가진다. 즉, 두 벡터의 차가 적으면 최적의 움직임 벡터가 주변에 있을 확률이 높으며 좁은 탐색 영역에서의 탐색으로도 효율적인 움직임 추정을 수행할 수 있다. 또한, 수신된 움직임 벡터와 예측 움직임 벡터의 차가 크다는 것은 해당 블록의 두 벡터와 움직임이 다를 확률이 높다는 것으로 이러한 경우 넓은 탐색 영역에서의 탐색으로 부호화 효율을 증가시킨다.

기존 H.264/AVC 등의 코덱에서는 모든 예측 모드의 윌-왜곡 값을 측정하여 최적의 모드를 결정하기 때문에 부호화 효율이 높지만, 복잡도 또한 높다. 하지만 제안한 방법은 프레임마다 할당된 복잡도 내에서 최적의 부호화를 수행하기 때문에 복잡도가 낮아도 부호화 효율을 높일 수 있다. 제안한 방법은 블록의 복잡도를 할당하기 이전에 블록의 스킵모드를 윌-왜곡 값을 이용하여 결정하고 스킵된 블록들을 제외한 나머지 블록들에 대하여 복잡도를 할당한다.

표 1. 스킵된 블록의 비율  
Table 1. Rate of skip block

시퀀스	QP	스킵 비율(%)
Akko	29	69
	33	76
	37	81
	41	85
Flamenco2	29	60
	33	65
	37	71
	41	79
Exit	29	88
	33	91
	37	93
	41	95
Rena	29	73
	33	79
	37	85
	41	89
평균		80

표 1은 복호화기에서 생성한 보조정보와 측정된 움직임 벡터를 기반으로 복원을 했을 경우 스킵될 확률을 나타내는 표로, 스킵 비율이 80%로 나타난다. 스킵으로 결정된 블록은 움직임 탐색을 수행하지 않기 때문에 프레임에 할당된 복잡도에서 스킵모드로 부호화 될 블록을 제외한 남은 블록들에게 복잡도를 더 많이 할당할 수 있다.

스킵모드의 빠른 결정을 위하여 스킵모드에 대한 윌-왜곡 값을 측정하고 이를 바탕으로 우선적으로 스킵이 수행될 블록을 결정한다. 아래의 식 (2)를 이용하여 스킵모드에 대한 윌-왜곡 값을 계산한다.

$$RDcost(skip) = Distortion + \lambda \cdot rate \quad (2)$$

*Distortion*은 예측 움직임 벡터가 가리키는 블록과 원본 블록의 왜곡값을 의미하며, *rate*는 움직임 벡터 차분치를 의미한다.  $\lambda$ 는 왜곡값과 움직임 벡터 차분치의 비율을 맞추기 위한 가중치를 나타낸다. 구해진  $RDcost(skip)$ 이 임계치( $TH\_skip$ )보다 작을 경우, 해당 블록은 스킵모드로 결정된다.

### 3. 블록 단위의 복잡도 할당

스킵모드로 선택이 되지 않은 블록들은 복호화기로부터 수신된 움직임 벡터와 부호화기에서 측정된 예측 움직임 벡터의 차이에 따라 복잡도를 할당하고 움직임 추정을 수행한다.

본 논문에서는 주변 블록을 참조하여 측정된 움직임 벡터와 복호화기에서 측정하여 전송한 움직임 벡터의 차이를

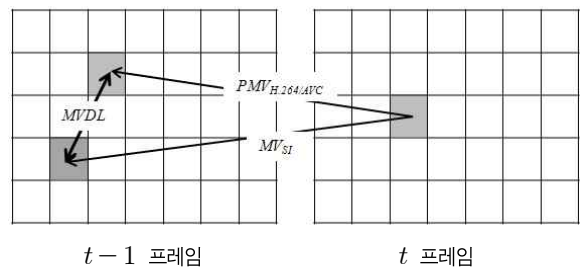


그림 5 . 제안한 방법의 MVDL  
Fig. 5. MVDL of the proposed method

$MVDL$ (Motion Vector Difference Level)라 정의한다. 그림 5는  $MVDL$ 을 구하는 방법을 도식화한 것이며, 다음의 식 (3)을 이용하여  $MVDL$ 를 계산한다.

$$MVDL = PMV_{H.264/AVC} - MV_{SI} \quad (3)$$

$PMV_{H.264/AVC}$ 는 주변 블록의 정보를 이용하여 측정된 예측 움직임 벡터를 의미하며,  $MV_{SI}$ 는 복호화기로부터 수신된 움직임 벡터를 의미한다. 측정된  $MVDL$ 의 크기에 따른 복잡도 분배는 다음의 식 (4)를 이용한다.

$$\omega = \begin{cases} w1, & \text{if } |MVDL| \leq L1 \\ w2, & \text{if } L1 < |MVDL| \leq L2 \\ w3, & \text{if } L2 < |MVDL| \leq L3 \\ w4, & \text{if } L3 < |MVDL| \end{cases} \quad (4)$$

$L1, L2$  그리고  $L3$ 은 임계치이며, 이에 따라 복잡도가 결정된다.

한편, 현재 프레임의 스킵 확률( $P(skip)$ )을 고려한 프레임 전체의 초기 복잡도( $C_i$ )는 다음과 같다.

$$C_i = C_t - B_{MB} \times P(skip) \quad (5)$$

식 (6)의  $C_t$ 는 프레임에 할당된 목표 복잡도를 나타내며,  $B_{MB}$ 는 프레임 내에 존재하는 블록의 수를 나타낸다. 스킵 모드는 예측 움직임 벡터가 가리키는 블록과의 SAD를 한번만 구한 뒤 결정되기 때문에 복잡도 1을 가진다. 따라서 스킵모드로 결정되고 남은 복잡도를 다른 블록에 할당하여 목표 복잡도 내에서 효과적인 움직임 추정을 수행할 수 있다.

복잡도를 분배하기 위하여, 현재 프레임에서 스킵이 아닌 블록의 개수( $B_i$ )는 다음의 식 (6)과 같이 측정된다.

$$B_i = B_{MB} \times (1 - P(skip)) \quad (6)$$

초기 복잡도와 스킵모드가 아닌 블록의 수가 결정되면 현재 블록의 복잡도  $C_n$ 은 식 (7)과 같이 결정된다.

$$C_n = w \times \frac{C_r}{B_r} \quad (7)$$

$C_r$ 은 초기 복잡도에서  $n-1$  번째 블록까지 스킵이 아닌 블록이 사용한 복잡도와 차이를 나타내며,  $B_r$ 은 초기 블록 개수와  $n-1$  번째까지 스킵이 아닌 블록 개수와의 차이값이다.

블록 단위로 할당된 복잡도에 따라 움직임 벡터 보정을 수행할 때, 제안한 방법에서는 복잡도를 기반으로 결정되는 가변 스텝 탐색을 사용한다. 가변 스텝 탐색은 탐색 스텝이 결정된 복잡도에 따라 아래와 같이 계산된다.

$$Step_n = \alpha w + \beta(k-1) \quad (8)$$

$\alpha$ 와  $\beta$ 는 가중치를 나타내며,  $k$ 는 복잡도의 레벨을 의미한다. 가변 스텝 탐색은 복잡도가 선형적으로 증가하는 것에 반하여, 검색 범위가 지수적으로 증가하기 때문에 실제 최적의 블록을 찾지 못할 가능성이 있지만 복잡도가 낮고 넓은 검색 범위를 가져 낮은 복잡도를 가지고 움직임이 많은 블록을 검색하는데 효율적이다.

#### IV. 실험 결과

제안한 방법의 성능 평가를 위해서, JVT Reference Software (version JM15.0)를 사용하였다. 실험 영상으로는 ‘Akko’, ‘Flamenco’, ‘Exit’와 ‘Rena’를 사용하였고, 각각 60 프레임씩 부호화하여 실험하였다. 양자화 계수는 37, 41, 45 그리고 49를 사용하였다. 또한, 보조정보를 생성하기 위하여 기존의 적응적 탐색 영역을 기반으로 보조정보를 생성하는 방법을 사용하였다<sup>[2]</sup>. 또한 기존 분산 비디오 코딩의 성능과 비교하기 위하여 수신된 움직임 벡터를 이용한 적응적 양자화 방법을 참고하였다<sup>[10]</sup>.

제안한 방법은  $16 \times 16$  크기의 블록을 사용하였으며, 블록의 스킵모드를 결정하는  $\lambda$ 는 H.264/AVC에서 이용하는 비율을 그대로 사용하였다. 스킵 모드를 결정하기 위한 임계치는 K.K. Ma의 논문을 이용하여 0.9로 정하였다<sup>[11]</sup>. 복



잡도를 결정하기 위한 임계치  $L1$ ,  $L2$  그리고  $L3$ 은 실험적으로 각각 13, 21과 29로 하였으며, 이에 따른  $w1$ ,  $w2$ ,  $w3$  그리고  $w4$ 는 1, 1.5, 2와 2.5로 결정하였다. 가변 스텝 탐색을 위한  $\alpha$ 와  $\beta$ , 역시, 실험을 통하여 결정된 값으로,  $\pm 1$ 의 탐색 영역에서 탐색을 수행할 수 있는 횟수가 9이기 때문에,  $\alpha$ 를 9로 실험하였다. 또한 각각의 스텝이 증가하면서 탐색 영역에서 탐색을 수행할 수 있는 횟수는 17, 25, 33 등으로 증가되기 때문에  $\beta$ 를 3.5로 결정하였다. 기존 H.264/AVC의 화면 간 단방향 예측 방법에서는  $\pm 32$  탐색영역으로 탐색하였다. 기존 분산 비디오 코딩 방법은 수신된 움직임 벡터 기반의 적응적 양자화 방법을 이용하였다<sup>[10]</sup>.

제안한 방법은 복호화기로부터 움직임 벡터를 전송받아 보정한 후 복호화기로 다시 움직임 벡터를 전송한다. 이때, 움직임 벡터는 H.264/AVC의 움직임 벡터 차분치(Motion Vector Difference)를 이용하며, 이를 엔트로피 코딩하여 복호화기로 전송한다. 표 2는 부호화기로부터 복호화기로 전송되는 움직임 벡터에 의한 추가 비트량을 나타낸 것이다. 일반적으로 QP가 높을 경우, 부호화 하는 블록이 스킵모드로 부호화 될 확률이 높고 스킵모드는 움직임 벡터를 보내

표 2. 움직임 벡터의 전송으로 인한 추가 비트량

Table 2 Bit rate according to motion vector transmission

영상	추가 비트량 (bps)	비트 증가율 (%)
Akko	1.81	0.29
flamenco	0.58	0.14
Exit	2.38	0.32
Rena	1.27	0.26
평균	1.51	0.25

지 않기 때문에 높은 QP에서 움직임 벡터의 부호화에 따른 비트량이 감소된다. 또한, 움직임이 적은 영상에 대해서 더 적은 비트가 발생됨을 확인할 수 있었다.

그림 6은 제안한 방법의 성능을 비교하기 위한 율-왜곡 곡선이다. 그림 6의 RD 그래프에서의 'IPIP'는 짝수 프레임은 H.264/AVC의 화면 내 예측 부호화되고 홀수 프레임은 H.264/AVC의 화면 간 단방향 부호화된 것을 의미하며, 'ALL I'는 모든 프레임을 H.264/AVC의 화면 내 부호화 방법으로 부호화 한 방법이다. 또한, 'r\_mv only'는 제안한 방법에서 복호화기로부터 부호화기로 움직임 벡터를 전송해 준 후에, 움직임 벡터 보정을 하지 않고 움직임 보상을 수행

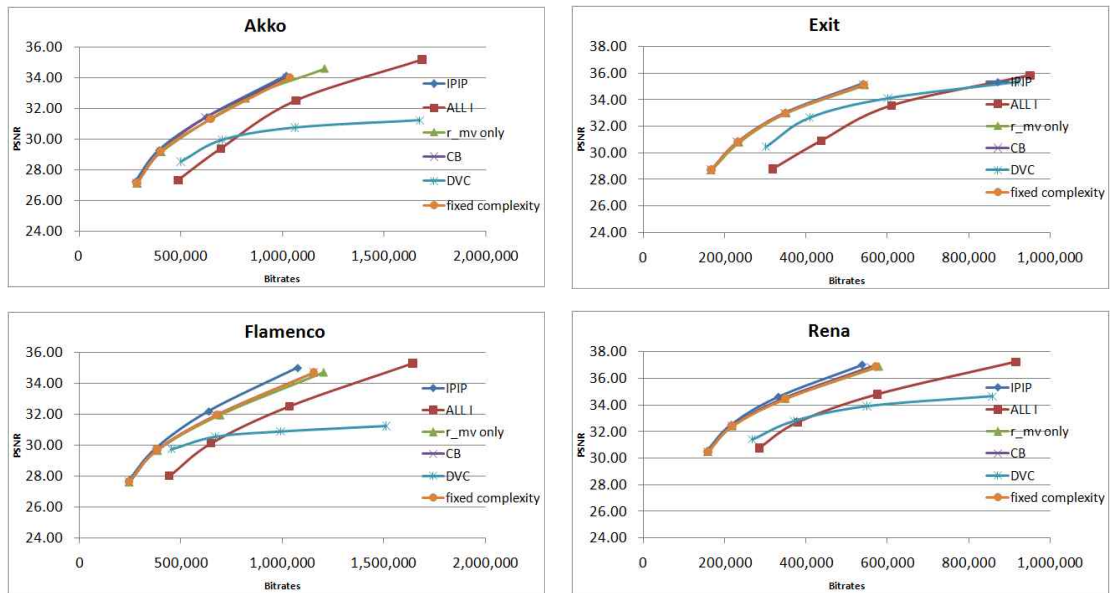


그림 6 . 제안한 방법과 기존 방법의 율-왜곡 곡선  
Fig. 6. RD curve of the proposed method and conventional method



하는 방법이고, ‘Fixed complexity’는 움직임 벡터의 보정은 수행하되 복잡도 분배를 수행하지 않는 방법을 나타낸다. ‘CB’는 복잡도 분배를 수행한 제안한 방법을 의미한다. 또한 ‘DVC’는 기존의 분산 비디오 코딩을 나타낸다. 표 3은 그림 6의 율-왜곡 그래프의 성능을 명확하게 수치로 표현한 것이며, 표 4는 각 부호화 방법에 따른 부호화기 복잡도를 나타낸다. 복잡도는 다음의 식 (9)로 계산된 값이다.

$$\frac{C_c - C_p}{C_c} \times 100 \tag{9}$$

$C_c$ 는 비교 대상의 복잡도를 의미하며  $C_p$ 는 제안한 ‘CB’ 방법의 복잡도를 의미한다.

제안한 ‘CB’방법은 ‘IPIP’방법보다 PSNR 성능이 0.17 dB 감소하였다. 기존의 ‘IPIP’방법보다는 가르침 비트가 추가되고 보조정보의 성능에 따른 추가 왜곡값 때문에 성능

저하가 있었다. 하지만 표 3에서 처럼 움직임 추정의 복잡도는 96%가 감소되었으며, GOP가 2로 부호화된 시퀀스의 전체 복잡도는 30% 감소한 것을 확인할 수 있었다. 화면 내 부호화 방법에서는 움직임 추정을 수행하지 않기 때문에 움직임 탐색의 감소된 비율을 적게 나타냈다. ‘All I’방법보다는 2.45dB의 성능향상이 있었다. ‘All I’방법에서는 수신된 움직임 벡터를 기반으로 부호화를 수행하였기 때문에 2.45dB가 증가하지만, 움직임 벡터를 보정하기 위한 움직임 추정 때문에 부호화 복잡도가 6% 증가하였다. 또한, 복잡도 분배를 하지 않고 부호화하는 ‘r\_mv\_only’방법보다 0.08dB 성능향상이 있었고, 복잡도는 3% 증가되었다. 이 방법은 움직임 추정을 수행하지 않기 때문에 움직임 추정의 복잡도가 0임으로 ∞의 복잡도를 가진다. 이는 정확한 보조정보의 움직임 벡터를 기반으로 움직임을 추정함으로써, 복잡도의 증가가 적었으며 일부 보조정보의 부정확한 움직임 벡터를 보완해줄 수 있었기 때문에 부호화 효율을

표 3 제안한 방법과 기존 방법의 율-왜곡  
Table 3 Rate-Distortion of the proposed method and conventional method

시퀀스	"IPIP"		"All I"		"r_mv_only"		"Fixed complexity"		"DVC"	
	BDpsnr (dB)	BDbitrate (%)	BDpsnr (dB)	BDbitrate (%)	BDpsnr (dB)	BDbitrate (%)	BDpsnr (dB)	BDbitrate (%)	BDpsnr (dB)	BDbitrate (%)
Akko	-0.07	1.26	2.52	-34.95	0.11	-2.17	0.15	-3.02	1.27	-26.45
Exit	-0.04	0.62	3.25	-41.77	0.05	-1.09	0.03	-0.59	1.27	-24.67
Flamenco	-0.34	7.57	1.71	-30.82	0.10	-2.34	0.02	-0.37	1.82	-32.55
Rena	-0.23	4.57	2.31	-37.92	0.06	-1.31	0.05	-1.04	2.18	-40.57
평균	-0.17	3.51	2.45	-36.36	0.08	-1.73	0.06	-1.25	1.63	-31.06

표 4. 제안한 방법과 기존 방법의 복잡도 비교  
Table 4. Comparison complexity of the proposed method and conventional method

시퀀스	"IPIP"		"All I"	"r_mv_only"		"Fixed complexity"		"DVC"
	움직임 추정 복잡도 (%)	복잡도 (%)	복잡도 (%)	움직임 추정 복잡도 (%)	복잡도 (%)	움직임 추정 복잡도 (%)	복잡도 (%)	복잡도 (%)
Akko	-96.38	-28.97	6.29	∞	1.63	0.50%	-0.53%	18.10
Exit	-96.89	-30.75	6.29	∞	1.55	-2.68%	-0.93%	18.11
Flamenco	-95.39	-25.89	7.14	∞	5.23	-2.73%	0.02%	19.04
Rena	-96.11	-28.04	5.17	∞	5.62	-0.67%	-0.24%	16.86
평균	-96.45	-30.35	6.27	∞	3.52	-1.40%	-0.24%	18.08

증가시킬 수 있었다. ‘Fixed complexity’방법 대비 성능은 0.06dB 향상되었으며 복잡도는 1%로 증가하였다. ‘DVC’방법과 대비해서는 1.63dB 성능향상이 있었으며, 부호화기 복잡도는 19%가 증가하였으나, 표 4에서처럼 복호화기의 복잡도는 99% 감소하였다. 기존 ‘DVC’방법은 부호화기에서 예측 및 움직임 추정을 수행하지 않기 때문에, 화면 내 예측 및 움직임 추정에 따른 복잡도가 증가되었다. 하지만, 채널 복호화기를 사용하지 않음으로써 복호화기 복잡도는 99% 감소시킬 수 있었다.

제안한 방법의 부호화기는 움직임 추정을 복호화기에서 수행하고 좁은 탐색영역에서 움직임 벡터를 보정하기 때문에 H.264/AVC의 부호화기의 복잡도보다 복잡도가 낮아졌고, 분산 비디오 부호화기의 부호화 성능보다는 부호화 성능이 증가됐다. 또한, 제안한 복호화기는 채널 복호화기 대신 엔트로피 코더를 사용하였기 때문에, 분산 비디오 복호화기의 복잡도보다 복잡도를 감소시킬 수 있었다.

표 5. 기존 분산 비디오 코딩과 제안한 방법의 복호화기 복잡도 비교  
Table 5. Complexity comparison between the conventional DVC decoder and the proposed decoder

시퀀스	복잡도 (%)
Akko	-99.9
flamenco	-99.9
Exit	-99.9
Rena	-99.9
Average	-99.9

## V. 결론

본 논문에서는 복호화기의 복잡도 감소를 위하여 엔트로피 코딩 기반의 분산 비디오 코딩을 제안하였으며, 목표 복잡도 안에서 부호화 효율을 높이기 위해 블록기반의 복잡도 적응적으로 분배하는 방법에 대하여 제안하였다. 실험 결과 기존의 분산 비디오 코딩에 비하여 부호화기의 복잡도는 약 11.8% 증가하였지만, 복호화기의 복잡도는 약

99% 감소 효과가 있었다. 또한, 목표 복잡도 내에서 블록 단위 복잡도 분배가 효율적으로 수행되는 것을 확인하였다. 하지만, 부호화기의 복잡도 증가와 백채널의 존재, 그리고 GOP의 크기가 고정되어있는 것은 여전히 해결해야 할 문제점으로 남아있다. 추후, GOP의 크기를 늘리면서도 복잡도를 효율적으로 관리할 수 있는 방법에 대하여 연구를 진행할 것이다.

## 참고 문헌

- [1] 심혁재, 전병우, “분산 비디오 압축 기술,” 전자공학회지, 제 36권 4호, 91-105쪽, 2009년 4월
- [2] K. Y. Min, S. N. Park and D. G. Sim, “Side information generation using adaptive search range for distributed video coding,” Conf. on RacRim 2009, pp. 854-857, Aug. 2009
- [3] W.J. Chien, L.J. Karam and G.P. Aboussleman, “Rate-distortion based selective decoding for pixel-domain distributed video coding,” Conf. on ICIP 2008, pp. 1132-1135, Oct. 2008
- [4] 박시내, 심동규, 전병우, “조명보상 기반 분산 다시점 비디오 코딩,” 전자공학회논문지, 제 45권, SP편, 제 6호, 17-26쪽, 2008년 11월
- [5] Draft ITU-T Recommendation and Final Draft international Standards of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC) Joint Video Team (JVT), Mar, Doc. 2003
- [6] T. Wiegand, G. J. Sullivan, G. Bjntegaard and A. Luthra, “Overview of the H.264/AVC video coding standard,” Trans. on IEEE Circuit and Systems for Video Technology, vol. 13, no. 7, pp. 560-576, July 2003
- [7] C. Brites and F. Pereira, “Encoder rate control for transform domain wyner-ziv video coding,” Conf. on ICIP 2007, vol. 2, pp. 5-8, Oct. 2007
- [8] J. L. Martinez, C. Holder, G. E. Fernandez, H. Kalva and F. Quiles, “DVC using a half-feedback based approach,” Conf. on ICME 2008, pp. 1125-1128, April 2008
- [9] C. Brites, J. Ascenso and F. Pereira, “Improving transform domain Wyner-Ziv video coding performance,” Conf. on ICASSP 2006, pp. 2-2, May 2006
- [10] 민경연, 박시내, 심동규, 김상호, “수신된 움직임 벡터를 이용한 적응적 블록 양자화 기반 분산 비디오 코딩 방법,” 한국통신학회논문지, 제 35권, 제 2호, 172-181쪽, 2010년 2월
- [11] H. Zeng, C. Cai and K. -K. Ma, “Fast mode decision for H.264/AVC based on macroblock motion activity,” Transactions on IEEE Circuits and System for Video Technology, vol. 19, no. 4, pp. 491-499, Apr. 2009

---

저 자 소 개

---



유 성 은

- 2011년 : 광운대학교 컴퓨터공학과 학사 과정
- 주관심분야 : 분산 비디오 코딩, 영상 처리



민 경 언

- 2009년 : 광운대학교 컴퓨터공학과 학사
- 2009년 : 광운대학교 컴퓨터공학과 석사 과정
- 주관심분야 : 분산 비디오 코딩, 화질 측정, 영상 처리



심 동 규

- 1999년 : 서강대학교 전자공학과 공학박사
- 2000년 : (주) 현대 전자
- 2002년 : (주) 바로 비전
- 2005년 : Univ. of Washington
- 현재 : 광운대학교 컴퓨터공학과 (부교수)
- 주관심분야 : 영상신호처리, 영상압축, 컴퓨터비전