

일반논문-11-16-1-11

보조정보의 움직임 벡터를 이용한 분산 비디오 코딩에서 H.264/AVC로의 트랜스코딩

민 경 연^{a)}, 유 성 은^{a)}, 심 동 규^{a)†}, 전 병 우^{b)}

Transcoding from Distributed Video Coding to H.264/AVC Based on Motion Vectors of Side Information

Kyung-Yeon Min^{a)}, Sung-Eun Yoo^{a)}, Dong-Gyu Sim^{a)†}, and Byeung-Woo Jeon^{b)}

요 약

본 논문은 저복잡도 및 고효율 분산 비디오 코딩에서 H.264/AVC로의 변환을 위한 트랜스코딩 방법을 제안한다. 제안한 방법은 낮은 복잡도로 높은 부호화 성능을 유지하기 위하여, 보조정보 생성을 위하여 측정된 움직임 벡터를 Wyner-Ziv (WZ) 프레임뿐만 아니라 키 프레임에서도 적용하여 부호화를 수행한다. 보조정보 생성을 위하여 측정된 움직임 벡터는 키 프레임에서 이전의 키 프레임으로의 움직임 추정에 의해 결정된 움직임 벡터임으로, 이 움직임 벡터를 이용하여 인트라 키 프레임을 예측 프레임으로 변환압축하는 방법을 제안한다. 또한, 제안한 방법은 두 예측 움직임 벡터를 기반으로 측정된 두 움직임 벡터 중, 비트율-왜곡 최적화를 수행하여 최적의 움직임 벡터를 선택한다. 보조정보의 움직임 벡터는 보조정보를 생성하기 위하여 수행된 움직임 추정을 통하여 측정된 움직임 벡터임으로, 적은 탐색 영역을 적용하여도 높은 부호화 효율을 얻을 수 있다. 따라서, 제안한 방법은 예측 움직임 벡터와 보조정보 기반의 예측 움직임 벡터로 적용하여 적은 탐색 영역에서 움직임 추정을 수행함으로써, 저복잡도로 높은 부호화 효율을 가질 수 있다. 실험 결과는 기존 변환 방법과 대비하여, 트랜스코더의 복잡도가 2.82%로 감소하고 비트율 성능은 23.06% 향상되었다.

Abstract

In this paper, a transcoding method with low computational complexity and high coding efficiency is proposed to transcode distributed video coding (DVC) bitstreams to H.264/AVC ones. For the proposed high-performance transcoding with low complexity, not only Wyner-Ziv frames but also key frames can be transcoded with motion vectors estimated in generation of side information. As a motion vector is estimated from a key frame to a prior key frame for side information generation, the motion vector can be used to encode the intra key frame as a predicted frame. Motion estimation is performed with two predicted motion vectors. One is the motion vector from side information generation and the other is median of motion vectors of neighboring blocks. The proposed method selects the best motion vector between two motion vectors based on rate-distortion optimization. Coding efficiency can be improved with a small size of search range, because a motion vector estimated in side information generation is used as an initial motion vector for transcoding. In the experimental results, complexity of transcoder is reduced about 12% and bitrate performance increases about 28.7%.

Keywords : 분산비디오부호화, H.264/AVC, 변환방법, 보조정보, 예측 움직임 벡터

1. 서론

최근 프로세서, 저장장치 및 전송기술이 급격하게 발달함에 따라, 멀티미디어 처리 기능을 가진 휴대 기기의 생산 및 보급이 활발하게 이루어졌다. 사용자는 휴대기기를 이용하여 멀티미디어 데이터를 취득하고, 콘텐츠를 제작 및 전송, 보급할 수 있게 된 것이다. 따라서, 제한된 연산 처리 능력 및 배터리 성능을 가진 환경에서의 사용을 위해, 짧은 부호화 시간과 낮은 전원소비가 가능한 부호기가 필요하게 되었다. 하지만, 기존의 표준 부호화 방법인 ISO MPEG-x 나 ITU-T H.26x 방법은 방송 서비스를 위해 연구, 개발된 기술로 하나의 서버에서 부호화한 데이터를 다수의 사용자에게 송신하는 스트리밍 방식에 최적화되어 설계되었다. 이에 따라, 복호기의 복잡도는 낮고 부호기의 복잡도가 높아, 복호화를 수행하는 동안에는 낮은 배터리 소비와 짧은 부호화 시간이 소요되지만, 부호화를 수행하는 동안 높은 배터리 소비와 긴 부호화 시간이 소요된다. 이러한 부호화 방법은 개개인의 콘텐츠 제작에서 이용되는 무선 센서 네트워크 장치에는 부적합하다^[1].

분산 비디오 부호화 방법은 Slepian-Wolf와 Wyner-Ziv의 정보이론을 기반으로 제안된 새로운 부호화 방법으로써, 부호기의 복잡도를 복호기로 분산시켜 부호기의 복잡도를 낮추는 부호화 방법이다. 따라서, 낮은 배터리 소비와 짧은 부호화 시간이 소요되어, 이동식 비디오 장치에 적합하다^[2]. 분산 비디오 부호화 방법은 입력 영상을 키 프레임과 WZ (Wyner-Ziv) 프레임으로 나눈 후, 키 프레임은 H.264/AVC 등 기존의 인트라 부호화 방식으로, WZ 프레임은 Wyner-Ziv 방식으로 부호화한다. Wyner-Ziv 부호화는 입력영상을 DCT (Discrete Cosine Transform) 변환과 양자화

한 후, Turbo 코드나 LDPC (Low-Density Parity-Check Accumulate) 코드 등으로 채널 부호화를 수행한다^[3-8]. 분산 비디오 복호화 방법은 키 프레임을 이용하여 보조정보를 생성하고, 이 보조정보와 전송 받은 패리티 비트를 채널 복호화를 수행한다. 보조정보를 생성하는 것과 채널 복호화를 수행하는 것은 높은 복잡도를 필요하므로, 분산 비디오 복호기의 복잡도가 상당히 높아, 이동식 비디오 장치에는 적합하지 않다.

따라서, 이동식 비디오에서 낮은 복잡도로 부호화 및 복호화를 수행하기 위하여, 분산 비디오 부호화 방법과 기존 H.264/AVC 비디오 복호화 방법의 장점을 이용한 변환 방법이 제안되고 있다. 이는, 단말기에서는 낮은 복잡도의 부호화와 복호화를 수행하고, 서버에서 높은 복잡도의 분산 비디오 복호기, 기존 비디오 부호기를 포함하여 수행한다. 기존의 분산 비디오 변환 방법은 크게 부호화 효율을 증가시키기 위한 방법과 서버의 지연을 감소하기 위한 방법으로 분류할 수 있다. 우선, 부호화 효율을 증가시키기 위한 방법으로는 분산 비디오 부호기에서 적용한 Group of picture (GOP의) 크기를 변환하여 재부호화를 수행하는 방법이다. 분산 비디오 부호기에서 기존의 화면 내 예측으로 부호화 되었던 키 프레임은 연장된 GOP의 구조에 맞추어 화면 간 예측으로 재부호화가 수행된다. WZ 프레임은 보조 정보에서 사용되었던 움직임 벡터를 이용하여 화면 간 예측 부호화 방법을 통하여 재부호화가 수행된다. 이 방법은 GOP의 크기를 증가시킴으로써 부호화 효율을 증가시킬 수 있으나, 서버에서의 지연이 증가한다는 문제점이 있다. 다음으로, 서버의 지연을 감소시키기 위한 방법으로는 분산 비디오 부호기에서 적용한 GOP의 크기를 그대로 적용하는 방법이다. 이 방법의 키 프레임은, 분산 비디오 부호기에서 이미 기존의 부호화 방법인 화면 내 예측을 이용하여 부호화되었기 때문에, 서버에서 복호화 후 재부호화를 수행하지 않고, 전송받은 비트스트림을 그대로 전송한다. 또한, WZ 프레임은 분산 비디오 복호기에서 WZ 프레임을 복호화하기 위하여 생성한 보조 정보의 움직임 벡터를 이용하여, 화면 간 예측을 이용하여 부호화를 수행한다. 이 방법은 화면 내 예측방법으로 부호화된 프레임을 재부호화를 수행하지 않으며 화면 간 예측 시, 움직임 추정의 복잡도를 감소

a) 광운대학교
Kwangwoon University

b) 성균관대학교
Sungkyunkwan University

✉ 교신저자 : 심동규 (dgsim@kw.ac.kr)

※ 본 연구는 지식경제부, 방송통신위원회 및 한국산업기술평가관리원의 산업원천기술개발사업(정보통신)의 일환으로 수행하였음. [K1002142, 차세대 모바일 영상서비스를 위한 초경량 비디오 부호화 원천기술개발 · 접수일(2010년10월29일), 수정일(2010년12월15일), 게재확정일(2010년12월20일)]

시킬 수 있기 때문에, 서버에서의 지연이 감소된다. 하지만, 보조정보의 성능 향상을 위하여 분산 비디오 부호화 방법에서 적용하는 GOP의 크기가 작기 때문에, 부호화 효율을 감소하게 된다는 단점이 있다.

제안하는 방법은 트랜스코딩의 지연을 감소시키면서 부호화 효율을 증가시키기 위하여, 보조정보의 움직임 벡터를 WZ 프레임을 화면 간 예측에서 사용할 뿐만 아니라 동시에 인트라 키 프레임을 화면 간 예측 프레임으로 부호화 시에도 움직임 벡터를 사용한다. 보조정보의 움직임 벡터는 보조정보 이 후의 키 프레임에서 이전의 키 프레임으로의 움직임 추정을 수행하여 결정됨으로써, 제안한 방법은 이 움직임 벡터를 키 프레임의 화면 간 예측 부호화에 적용한다. 따라서, 보조정보의 움직임 벡터를 WZ 프레임과 키 프레임의 화면 간 예측 부호화 시 적용하여 저복잡도로 높은 부호화 효율을 얻는다. 또한, 제안한 방법은 보조정보 기반의 예측 움직임 벡터와 주변 블록 정보 기반의 예측 움직임 벡터를 중심으로 적은 탐색 영역에서 움직임 추정을 수행한다. 각각의 탐색 영역에서 측정된 두개의 움직임 벡터의 비트율-왜곡 최적화를 수행하여 최적의 벡터를 선택한다. 적은 탐색 영역에서의 움직임 추정을 수행하여도, 보조정보 기반의 예측 움직임 벡터는 움직임 추정을 통하여 측정된 값임으로 움직임 추정의 신뢰도를 높일 수 있다. 따라서, 제안한 방법의 트랜스코딩 방법은 저복잡도로

GOP의 크기를 증가시켜 높은 부호화 효율을 얻을 수 있다.

본 논문의 구성은 다음과 같다. II장에서는 기존 연구에 대하여 살펴보고, III장에서는 제안하는 알고리즘에 대하여 자세히 소개한다. IV장에서는 제안하는 알고리즘에 대한 실험 방법과 실험 결과 및 그 결과에 대한 분석을 기술하고, 마지막으로 V장에서는 이 논문의 결론 및 향후 연구계획에 대해 논한다.

II. 기존 방법

1. 분산 비디오 코딩

분산비디오 코딩은 Slepian-Wolf와 Wyner-Ziv의 정보이론 관점으로 부터 발전된 새로운 비디오 코딩 방법이다. 이 방법은 부호기의 높은 연산과정이 필요한 모듈을 복호기로 분배함으로써, 부호기의 복잡도를 감소시키면서 부호화 효율을 증가시키는 방법이다^[3-8]. 그림 1은 분산 비디오 코딩의 기본 블록도를 나타낸다. 분산 비디오 부호화 방법은 입력 영상을 키 프레임과 WZ (Wyner-Ziv) 프레임으로 구분한 후, 키 프레임은 H.264/AVC 등 기존의 인트라 부호화 방식으로, WZ 프레임은 채널 부호기를 이용하여 부호화한다. WZ 프레임은 DCT (Discrete Cosine Transform) 변환

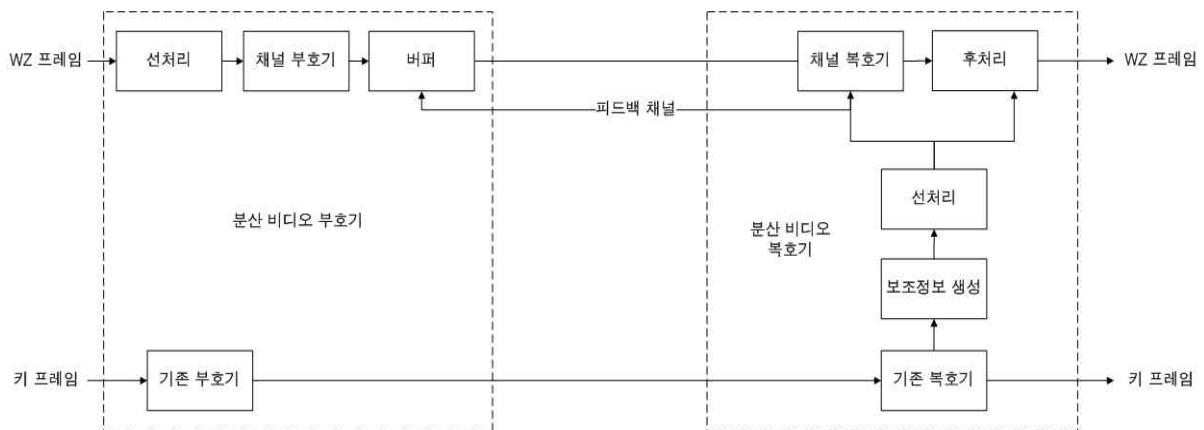


그림 1. 기본 분산 비디오 코딩의 블록도
Fig. 1. Block diagram of fundamental distributed video coding

과 양자화 과정의 선처리를 거친 후, Turbo 코드나 LDPCA (Low-Density Parity-Check Accumulate) 코드 등으로 채널 부호화를 수행한다^[3-8]. 채널 부호기에서 출력된 패리티 비트는 버퍼에 저장되어, 피드백 채널을 통하여 요청이 오면 패리티 비트를 순차적으로 전송하게 된다. 분산 비디오 복호기는 기존의 복호기를 이용하여 키 프레임을 복원하고, 복원된 키 프레임들을 이용하여 보조정보를 생성한다. 생성된 보조정보를 부호기에서 수행한 전처리와 같은 과정을 거친 후, 전송받은 패리티 비트를 이용하여 채널 복호화를 수행하여, 역양자화와 역DCT 등의 후처리를 거치게 되면 WZ 프레임이 복원한다.

기존 채널 부호기의 출력은 입력한 비트의 두 배가 출력으로 나오게 되는데, 부호화 효율을 위하여 원본 영상에 관한 비트를 제외하고 패리티 비트만을 전송한다. Stanford 대학에서 제공하는 LDPCA는 패리티비트를 66단계로 나누어, 피드백 채널을 통하여 요청이 입력될 때마다 단계 별로 전송한다^[9]. 이 때, 66단계까지 전송하게 되면 입력영상과 같은 양의 비트를 전송하는 것과 같아진다. 분산 비디오 복호기는 전송받은 패리티 비트와 보조정보를 이용하여 복

원하게 되는데, 보조정보란 WZ의 원본 영상과 유사하게 생성된 영상을 의미한다. 이 보조정보와 원본영상의 차를 채널을 통한 오류로 가정하여, 패리티 비트를 이용하여 복원한다. GOP의 크기가 증가되면, 보조정보의 성능이 급격히 하락하기 때문에, 분산 비디오 코딩에서는 일반적으로 GOP의 크기를 두 프레임으로 부호화를 수행한다. 분산 비디오 코딩의 성능은 보조정보의 성능과 밀접한 관계를 가지기 때문에, 높은 성능의 보조정보를 생성하기 위한 방법들이 활발히 제안되고 있다.

2. 분산 비디오 코딩의 보조정보

보조정보의 성능은 분산 비디오 부호화 방법의 성능과 긴밀한 연관성을 가지고 있기 때문에, 보조정보의 성능 향상을 위한 많은 방법들이 제안되고 있다. 보조정보를 생성하는 방법은 크게 움직임을 고려하지 않은 방법과 움직임 기반의 알고리즘으로 나눌 수 있다. 하지만, 움직임을 고려하지 않은 방법은 움직임 기반의 알고리즘보다 대체적으로 성능이 낮기 때문에, 대체적으로 움직임을 고려한 알고리

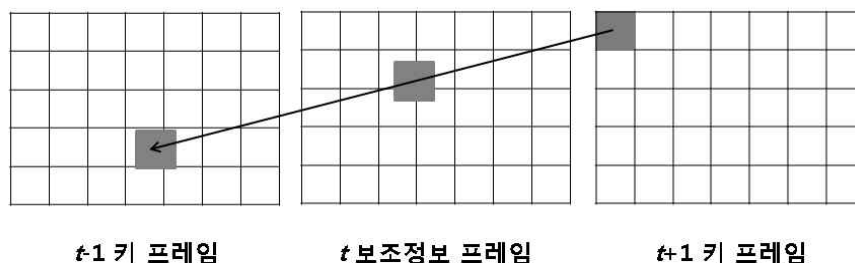


그림 2. 단방향 움직임 추정 기반의 보조정보 생성
Fig. 2. Side information generation based on unidirectional motion estimation

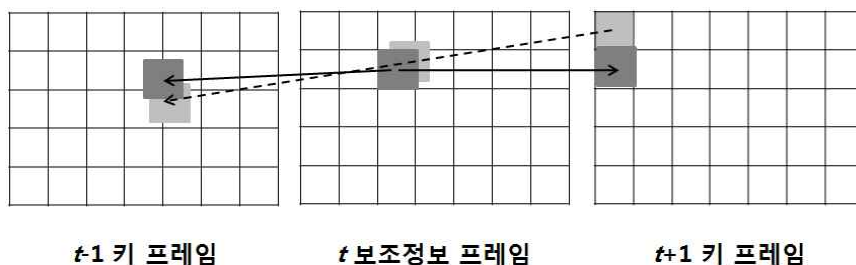


그림 3. 양방향 움직임 추정 기반의 보조정보 생성
Fig. 3. Side information generation based on bidirectional motion estimation

즘이 제안되고 있다. 움직임 기반의 알고리즘은 또다시, 단일 움직임 추정과 양방향 움직임 추정을 기반으로 하는 알고리즘이 있다. 우선, 단일 움직임 추정 방법의 보조정보 생성 알고리즘은 (t+1) 시간의 키 프레임에서 (t-1) 시간의 키 프레임으로 움직임 추정을 수행하고, 측정된 움직임 벡터의 중간 위치에 해당 블록을 보간하여 보조정보를 생성하는 것이다. 따라서 그림 2처럼 (t+1) 프레임을 기준으로 움직임 벡터가 측정된다. 다음으로 양방향 움직임 추정 기법을 기반으로 한 보조정보 생성 방법은 (t+1) 프레임에서 (t-1) 프레임으로 움직임 추정을 수행하고, 추정된 움직임 벡터를 바탕으로 t 프레임을 기준으로 (t-1) 프레임과 (t+1) 프레임으로 움직임 추정을 한 번 더 수행한다. 따라서, 그림 3처럼 t 프레임을 기준으로 움직임 벡터가 측정된다.

3. 분산 비디오 코딩에서 H.264/AVC로의 기존 트랜스코딩 방법

이동식 장치의 부담을 감소하기 위한 분산 비디오 코딩에서 H.264/AVC로의 변환 방법은 크게 부호화 효율을 최

대화하기 위한 방법과 서버의 지연을 감소시키기 위한 방법으로 분류할 수 있다^[12-13].

우선, 부호화 효율을 최대화하기 위한 방법은 서버에서 분산 비디오 복호화를 수행하고, 복원된 프레임들을 이용하여 최대한의 부호화 효율을 나타낼 수 있도록 부호화를 수행하는 것이다. 부호화 효율을 위하여, 일반적으로 분산 비디오 코딩에서는 GOP의 크기를 두 프레임으로 부호화를 수행한다. 하지만, H.264/AVC에서는 GOP의 크기가 작아지면 화면 내 예측 방법으로 부호화된 프레임의 주기가 증가되면서 압축효율이 감소하게 된다. 따라서, 화면 내 예측 방법으로 부호화된 프레임을 화면 간 예측으로 변경함으로써, 부호화 효율을 증가시킨다. 이러한 방법들은 보조정보의 움직임 벡터를 WZ 프레임으로 화면 간 예측 부호화를 수행하기 위하여 이용하지만, 키 프레임의 화면 간 예측 부호화를 위해서는 움직임 예측을 수행한다. 이러한 방법은 GOP의 크기를 증가시킴으로써 부호화 효율을 증가시킬 수 있다. 하지만, 보조정보의 움직임 벡터를 그대로 WZ 프레임의 화면 간 예측 부호화를 위한 움직임 벡터로 적용하면 움직임 벡터 차분값 (MVD)가 증가되어, 기존의 H.264/

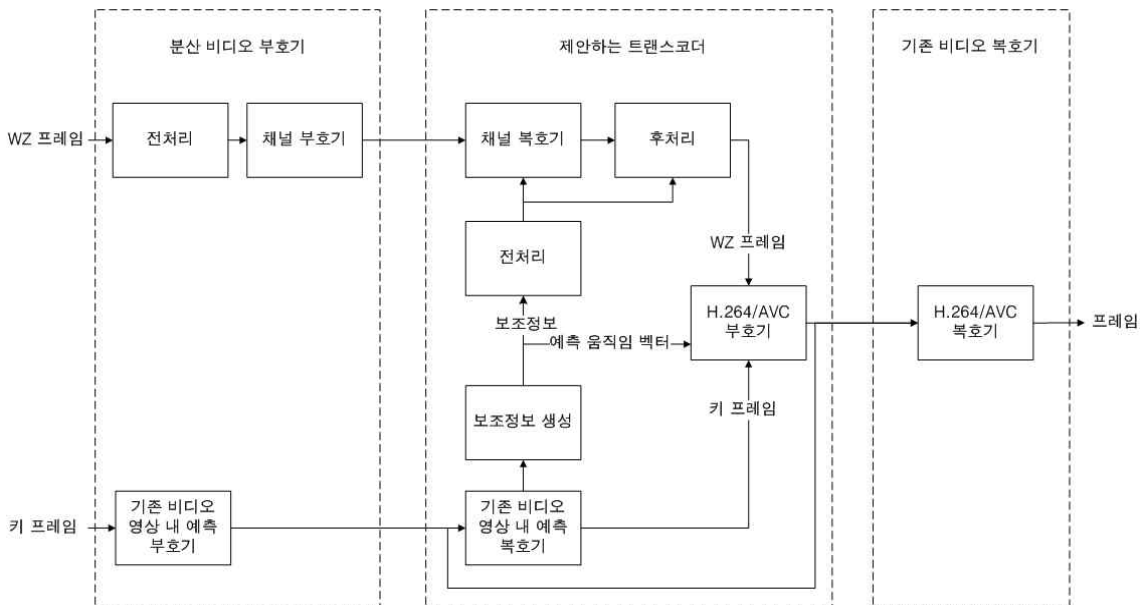


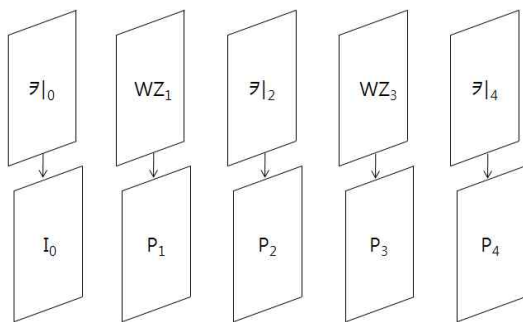
그림 4. 제안하는 방법의 블록도
Fig. 4. Proposed block-diagram

AVC 화면 간 예측 부호화 방법과 비교하여 부호화 효율이 감소될 수 있다. 또한, 키 프레임의 트랜스코딩을 위한 넓은 영역에서의 움직임 추정 때문에 서버에서의 지연이 증가한다는 문제점이 있다.

다음으로, 서버의 지연을 감소시키기 위한 방법으로는 분산 비디오 부호기에서 적용한 GOP의 크기를 그대로 적용하는 방법이다^[12-13]. 이 방법의 키 프레임은, 분산 비디오 부호기에서 이미 기존의 H.264/AVC 부호화 방법인 화면 내 예측을 이용하여 부호화가 수행되었기 때문에, 서버에서 복호화 후 재부호화를 수행하지 않고, 전송받은 비트스트림을 그대로 전송한다. 또한, WZ 프레임은 분산 비디오 복호기에서 WZ 프레임을 복호화하기 위하여 생성한 보조정보의 움직임 벡터를 이용하여, 화면 간 예측을 이용하여 부호화를 수행한다. 이러한 방법은 화면 내 예측방법으로 부호화된 프레임을 재부호화를 수행하지 않으며 화면 간 예측 시, 보조정보의 움직임 벡터를 이용함으로써 움직임 추정의 복잡도를 감소시킬 수 있기 때문에, 서버에서의 지연이 감소된다. 하지만, 보조정보의 성능 향상을 위하여 분산 비디오 부호화 방법에서 적용하는 GOP의 크기가 작기 때문에, 트랜스코딩 후에도 기존의 H.264/AVC의 부호화 효율이 감소된다.

III. 제안하는 방법

저 사양의 이동식 비디오 장치에 적합한 저복잡도 부호

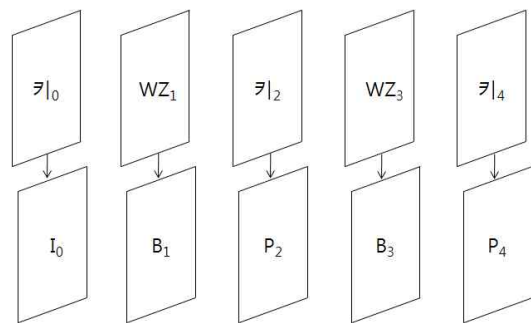


(a) 분산 비디오 코딩으로 부호화된 프레임을 "IPP" 방법의 트랜스코딩

기와 복호기를 위하여, 분산 비디오 부호기와 H.264/AVC 비디오 복호기를 접목하기 위한 변환 방법이 요구된다. 서버의 지연이 적으면서도 높은 부호화 효율의 트랜스코딩을 위하여, 제안한 방법은 보조정보의 움직임 벡터를 이용하여 키 프레임과 WZ 프레임의 화면 간 예측 부호화를 수행하며, 그림 4는 제안한 방법의 블록도를 나타낸다. WZ 프레임과 키 프레임의 비트스트림이 서버로 입력되면, 트랜스코더는 이를 복호화하고, WZ 프레임의 복호화를 위해서 필수적으로 요구되는 보조정보를 생성한다. 제안한 트랜스코딩 방법은 보조정보 생성 시, 측정된 움직임 벡터와 H.264/AVC의 예측 움직임 벡터를 이용하여 WZ 프레임과 키 프레임을 화면 간 예측 방법으로 부호화를 수행한다. 또한, GOP의 구조에 의하여 화면 내 예측 방법으로 부호화된 프레임이 필요할 경우, 트랜스코더에 입력된 키 프레임의 비트스트림을 그대로 전송하도록 한하여 복잡도를 감소시키도록 한다. 따라서, 제안한 방법은 보조정보의 움직임 벡터를 WZ 프레임뿐만 아니라 키 프레임에서도 이용함으로써, 저복잡도로 GOP의 크기를 증가시킴으로써, 부호화 효율을 증가시킬 수 있다.

1. GOP의 크기 변환 및 보조정보의 움직임 추정에 따른 부호화 방법

분산 비디오 코딩의 효율은 보조정보의 성능과 긴밀한 관계를 가지기 때문에, 일반적으로 GOP의 크기를 2로 부



(b) 분산 비디오 코딩으로 부호화된 프레임을 "IBP" 방법의 트랜스코딩

그림 5. 예측 모드에 따른 트랜스코딩
Fig. 5 Transcoding according to prediction mode

호화를 수행한다. 하지만 H.264/AVC에서 GOP의 크기가 작으면, 화면 내 예측방법으로 부호화된 프레임의 분포가 높아 오류에는 강하나 부호화 효율이 크게 감소하게 되기 때문에, 일반적으로 15 프레임마다 화면 내 예측방법으로 부호화된 프레임을 삽입한다. 따라서, 제안한 방법은 보조정보의 움직임 벡터를 적용하여, 저복잡도로 그림 5와 같이 GOP 크기를 변환하는 트랜스코딩을 수행한다. 제안한 방법은 트랜스코더의 프로파일, 성능 및 상태에 따라 화면 간 단방향과 양방향 예측을 수행할 수 있다. 우선, 그림 5의 (a)와 같이 반복적인 키 프레임과 WZ 프레임들을 첫 프레임은 화면 내 예측 방법으로 나머지의 프레임들은 화면 간 단방향 예측 방법으로 부호화는 트랜스코딩 방법을 나타내며, 이를 간략하게 표현하기 위하여 “IPP”방법이라 표현한다고 가정한다. 또한, 첫 프레임은 화면 내 예측 방법으로 나머지 프레임들은 화면 간 단방향 예측 방법과 양방향 예측 방법으로 번갈아가면서 부호화를 수행하는 트랜스코딩 방법을 “IBP” 방법이라고 명칭한다.

지연 감소를 위하여 제안한 방법은 t 시간의 보조정보를 생성하기 위하여 측정된 움직임 벡터를 t 시간의 프레임과 (t+1) 시간의 프레임에서 공유하여 사용하며, 압축효율을 증가시키기 위하여 보조정보의 움직임 추정 방식에 따라 화면 간 단방향 예측 부호화와 화면 간 양방향 예측 부호화 방법이 다르게 수행한다. 그림 6은 움직임 추정 방법과 예측 모드에 따라 t 프레임과 t+1 프레임의 H.264/AVC 기반의 예측 움직임 벡터, 참조 프레임과 보조정보 기반의 예측 움직임 벡터를 결정하기 위한 흐름도를 나타낸다. 또한, 그림 7과 그림 8은 그림 6의 흐름도를 도식화한 것이다. 그림 6, 그림 7과 그림 8에서의 PMV_{H.264/AVC,t}와 PMV_{H.264/AVC,t+1}는 각각 t 프레임과 t+1 프레임의 H.264/AVC 기반의 예측 움직임 벡터를 의미한다. 이 H.264/AVC 기반의 예측 움직임 벡터는 주변 블록들의 움직임 벡터의 중간값으로 결정되는 기존 H.264/AVC에서 예측 움직임 벡터를 측정하는 방법으로 결정되어진다. PMV_{SI,t}와 PMV_{SI,t+1}은 각각 t 프레임과 t+1 프레임의 보조정보 기반의 움직임 벡

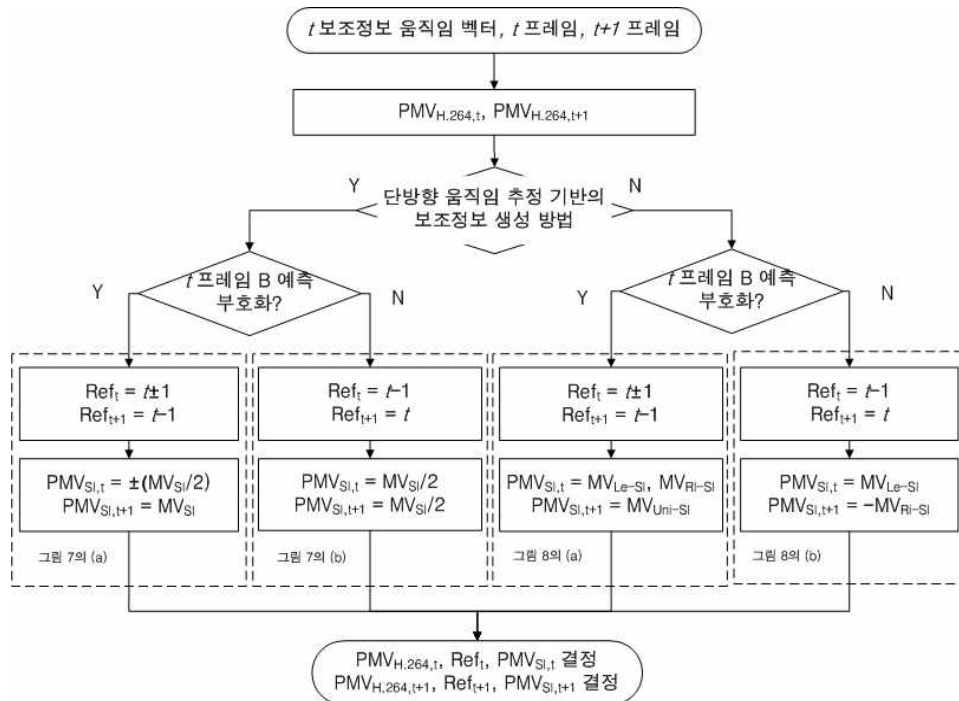


그림 6. 보조정보의 움직임 추정에 따른 예측 부호화 방법
 Fig. 6. Prediction coding method according to motion estimation of side information

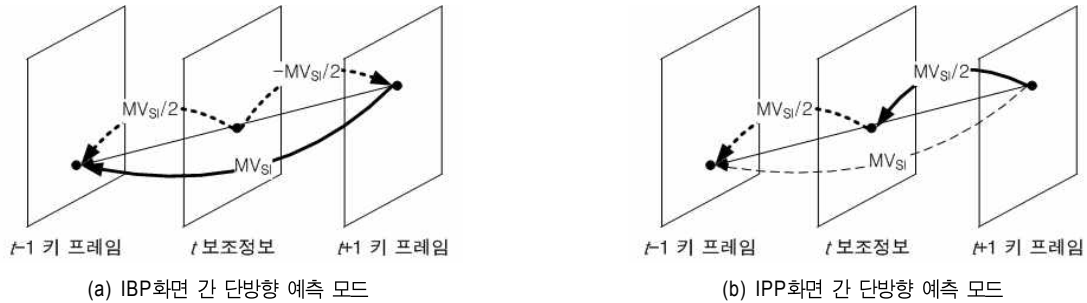


그림 7. 단방향 움직임 추정 기반의 보조정보 생성 시 예측 부호화 방법
 Fig. 7. Prediction coding method according to side information generation based on unidirectional motion estimation

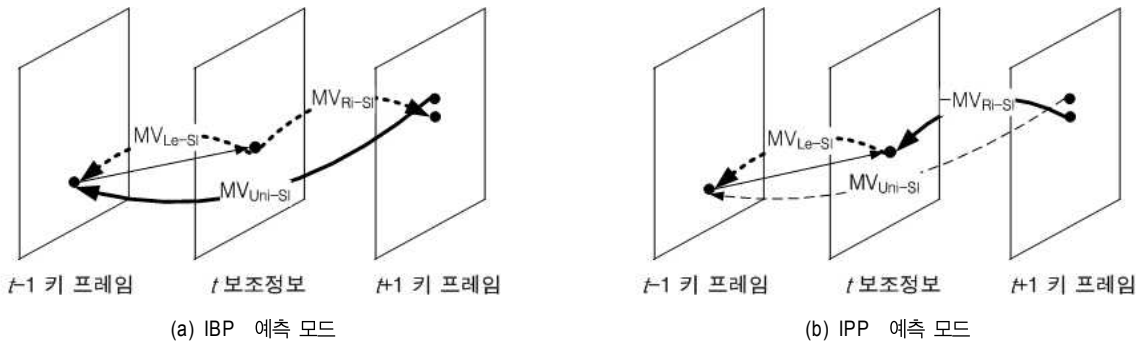


그림 8. 양방향 움직임 추정 기반의 보조정보 생성 시 부호화 방법
 Fig. 8. Prediction coding method according to side information generation based on bidirectional motion estimation

터를 의미한다. 이 보조정보 기반의 움직임 벡터는 수신된 보조정보의 움직임 벡터를 의미한다. MV_{SI} 는 단방향 움직임 예측을 통하여 측정된 $t+1$ 키 프레임에서 $t-1$ 키 프레임으로의 움직임 벡터이다. 또한, MV_{Uni-SI} 는 양방향 움직임 예측을 위하여 측정된 $t+1$ 키 프레임에서 $t-1$ 키 프레임으로의 단방향 움직임 벡터, MV_{Le-SI} 는 보조정보를 중심으로 $t-1$ 프레임의 움직임 벡터를 의미하며, MV_{Ri-SI} 는 보조정보를 중심으로 $t+1$ 프레임의 움직임 벡터를 의미한다. Ref_t 와 Ref_{t+1} 는 각각 t 프레임에서 참조될 프레임과 $t+1$ 프레임에서 참조될 프레임을 의미한다. 우선, t 보조정보의 움직임 벡터, t 프레임과 $(t+1)$ 프레임이 입력되면, 주변 블록의 움직임 벡터를 이용한 H.264/AVC의 기존 예측 움직임 벡터를 추정한다. 입력된 보조정보의 움직임 벡터가 단방향 움직임 추정 기반으로 생성된 보조정보의 움직임 벡터를 이용하고, t 프레임과 $(t+1)$ 프레임을 각각 화면 간 양방향 예측 방법과 화면 간 단방향 예측 방법으로 부호

화는 경우를 도식화하여 표현한 것이 그림 7의 (a)로, t 프레임은 $(t-1)$ 프레임과 $(t+1)$ 프레임을 참조 프레임 (Ref_t) 으로 하며, 보조정보 기반의 예측 움직임 벡터 ($PMV_{SI,t}$) 를 보조정보 움직임 벡터 (MV_{SI}) 의 $\pm 1/2$ 값을 이용한다. $(t+1)$ 프레임의 참조 프레임은 $(t-1)$ 프레임 (Ref_{t+1}) 을 이용하고 보조정보의 움직임 벡터 (MV_{SI}) 를 그대로 보조정보 기반의 예측 움직임 벡터 ($PMV_{SI,t+1}$) 로 이용한다. t 프레임 역시 화면 간 단방향 예측을 수행할 경우를 도식화하여 표현한 것이 그림 7의 (b)로, t 프레임은 $(t-1)$ 프레임을 참조 프레임 (Ref_t) 으로, $t+1$ 프레임은 t 프레임을 참조 프레임 (Ref_{t+1}) 으로 하고, 두 프레임에서 모두 보조정보 기반의 예측 움직임 벡터 ($PMV_{SI,t}$ 와 $PMV_{SI,t+1}$) 를 보조정보의 움직임 벡터 (MV_{SI}) 의 $1/2$ 값을 이용한다. 입력된 움직임 벡터가 양방향 움직임 추정 기반으로 측정되고, t 프레임을 화면 간 양방향 예측 부호화를 수행할 경우를 도식화한 것이 그림 8의 (a)로, 참조 프레임 (Ref_t) 은 $(t-1)$

프레임과 (t+1) 프레임으로 하며, 양방향 움직임 벡터 (MVLE-SI와 MVRi-SI) 를 보조정보 기반의 예측 움직임 벡터 (PMVSI,t) 로 하여 수행한다. (t+1) 프레임은 (t-1) 프레임을 참조 프레임 (Ref_t+1) 으로 하며 보조정보 기반의 예측 움직임 벡터 (PMVSI,t+1) 는 단방향 보조정보 움직임 벡터 (MVUni-SI) 를 그대로 사용한다. t 프레임 역시 화면 간 단방향 예측을 수행할 경우를 도식화한 것이 그림 8의 (b)로, t 프레임의 참조 프레임 (Ref_t) 을 (t-1) 프레임으로 적용하고 보조정보 기반의 예측 움직임 벡터 (PMVSI,t) 는 양방향 움직임 벡터 (MVLE-SI) 를 이용하고, t+1 프레임의 참조 프레임 (Ref_t+1) 을 t 프레임, 보조정보 기반의 예측 움직임 벡터 (PMVSI,t+1) 는 양방향 움직임 벡터 (-MVRi-SI) 를 적용한다.

2. 탐색 영역 결정

넓은 영역에서의 탐색은 더 유사한 블록을 찾을 수 있는 확률이 증가하나, 복잡도 역시 증가하는 반면, 적은 영역에서의 탐색은 복잡도는 감소하지만 부호화 효율 역시 감소된다. 하지만 제안하는 방법은 H.264/AVC의 예측 움직임 벡터와 보조정보 기반의 예측 움직임 벡터를 이용하기 때문에, 부호화의 효율에 손해가 거의 없이 적은 범위의 탐색으로 복잡도를 감소시킬 수 있다.

그림 9는 제안한 방법의 탐색 영역과 기존 H.264/AVC의 탐색 영역을 나타낸 것으로, 기존의 움직임 추정을 수행하기 위하여 예측 움직임 벡터를 중심으로 한 탐색 범위가 점선 영역 (c) 일 때, 제안한 방법의 탐색 영역은 예측 움직임 벡터를 중심으로 한 회색 영역 (b) 과 보조정보 기반의 예측 움직임 벡터를 중심으로 한 회색 영역 (a)이 된다. 따라서 각각의 회색 영역에서만 움직임 추정을 수행한다.

보조정보의 생성을 위하여 움직임 추정을 통해 이미 최적의 움직임 벡터를 측정하였기 때문에, 보조정보 기반의 예측 움직임 벡터를 중심으로 적은 영역에서 탐색을 수행하여도 부호화 효율이 크게 감소되지 않는 것이 가능하다. 하지만, 선택된 블록과의 차가 적어도, 예측 움직임 벡터와 움직임 벡터 간의 차이가 매우 커지면 비트율이 증가하게 된다. 따라서, 제안한 방법은 예측 움직임 벡터를 중심에서

도 최적의 블록을 탐색하여, 두 영역에서 선택된 블록 중 하나의 블록을 선택하게 된다.

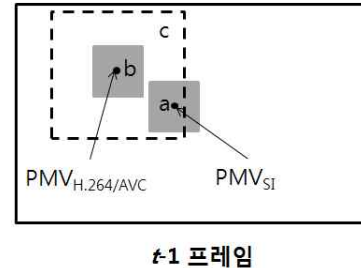


그림 9. 탐색 영역 결정
Fig. 9. Decision of search range

3. 움직임 벡터 결정

H.264/AVC의 표준은 블록의 움직임 벡터를 직접 보내는 것이 아니라, 주변 블록을 이용하여 예측 움직임 벡터를 측정하고, 결정된 움직임 벡터와 예측 움직임 벡터와의 차분치를 전송한다. 따라서 결정된 블록과의 차분치 뿐만 아니라 움직임 벡터 차가 적은 블록이 최적의 부호화 효율을 나타낼 수 있다.

제안한 방법은 그림 9의 각각의 회색 영역에서 선택된 두 블록 중 하나의 블록을 선택하기 위하여 비트율-왜곡 최적화를 식 (1)을 수행하고, 최적의 블록을 결정한다.

$$MV_{BEST} = \underset{i \in a, b}{\operatorname{argmin}} (\lambda MV_{cost}(i) + distortion(i)), \quad (1)$$

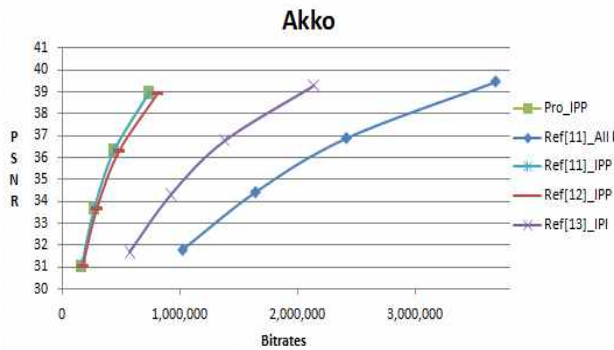
λ 는 가중치를 나타낸다. 최적의 움직임 벡터는 움직임 벡터 차분값과 잔여부분을 부호화하는 비트와 화질을 함께 고려하여 결정된다. 제안하는 방법은 기존 H.264/AVC의 비트율-왜곡 최적화에서 그림 9의 (a) 영역에서 구한 블록과 (b) 영역에서 구한 블록을 함께 고려하여 최적의 움직임 벡터를 결정한다. 따라서, 제안한 방법은 H.264/AVC의 표준을 따르면서, 저복잡도로 최적의 움직임 벡터를 측정할 수 있다.

IV. 실험 결과

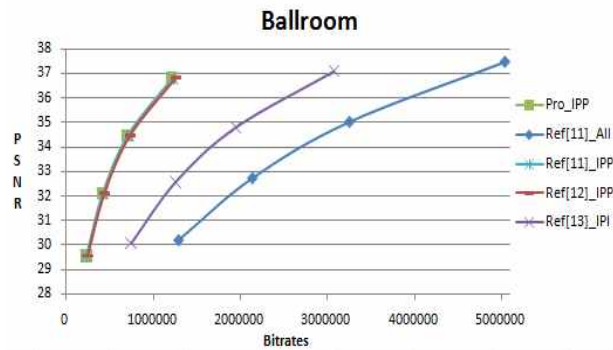
제안한 방법의 성능을 평가하기 위해 기존 방법과 비트의 사용율과 PSNR을 측정하였으며, 변환코더의 복잡도를 측정하였다. 실험은 640×480 크기의 VGA급 영상, “Akko”, “Ballroom”, “Exit”와 “Flamenco2”를 이용하였으며, 각각 4 개씩의 QP 값을 이용하였다. 각각의 영상을 각각의 QP에 따라 250장을 이용하여 실험을 수행하였으며, 보조정보 프레임을 생성하는 방법은 참조 논문[6] 방법을 이용하였다. 또한, H.264/AVC를 위하여 JM15.0로 실험을 진행하였다. 제안한 방법 중, 보조정보 기반의 예측 움직임 벡터를 기준으로 한 탐색 영역은 ±16 크기로 설정하였고, H.264/AVC의 예측 움직임 벡터를 중심으로 ±8 크기의 탐색 영역에서 움직임 추정을 수행하도록 하였다. 기존

H.264/AVC의 화면 간 예측의 탐색 영역은 ±32 크기로 하여 실험하였다.

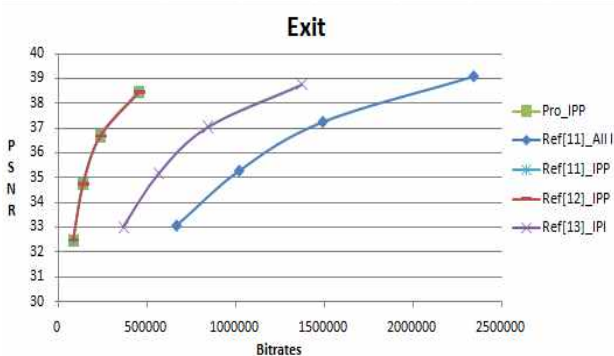
그림 10은 화면 간 단방향 참조 변환 비트율-왜곡 그래프로 기존 트랜스코딩 방법과 제안한 방법의 성능을 나타낸다. 그림 10의 “Pro_IPP”는 제안한 방법의 RD 성능을 나타내며, “All I”는 모든 프레임을 H.264/AVC의 화면 내 예측 방법으로 부호화를 수행한 기존 방법의 결과이다^[11]. “IPP”로 표시된 RD 그래프는 첫 프레임은 H.264/AVC의 화면 내 예측 방법으로 부호화를 수행하고 나머지 프레임들은 H.264/AVC의 화면 간 단방향 예측을 수행하는 기존의 방법이다^[12]. “IP”는 기존의 방법으로 GOP 크기가 2로 짝수 프레임은 화면 내 예측 방법으로 부호화를 수행하고, 홀수 프레임은 화면 간 단방향 예측 방법으로 부호화를 수행한 것이다^[13]. 제안한 방법은 H.264/AVC의 “IP” 방법보다는



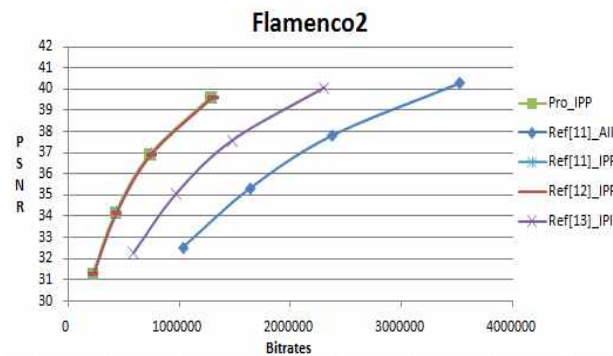
(a) “Akko” 비트율-왜곡 그래프



(b) “Ballroom” 비트율-왜곡 그래프

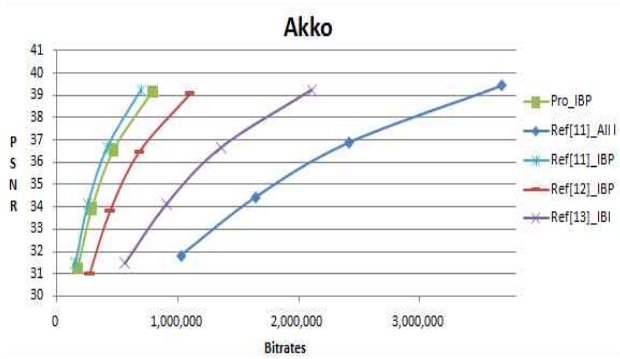


(c) “Exit” 비트율-왜곡 그래프

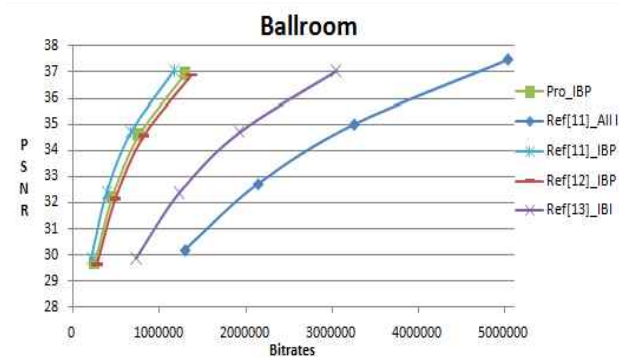


(d) “Flamenco2” 비트율-왜곡 그래프

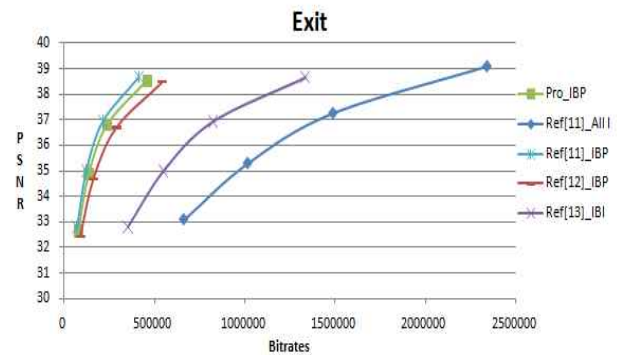
그림 10. “IPP” 변환 비트율-왜곡 그래프
Fig. 10. RD graph of “IPP” transcoding



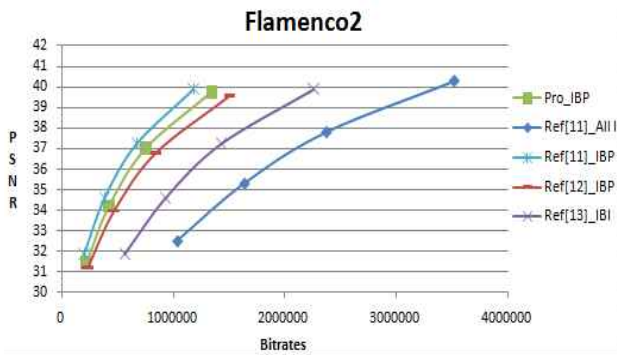
(a) "Akko" 비트율-왜곡 그래프



(b) "Ballroom" 비트율-왜곡 그래프



(c) "Exit" 비트율-왜곡 그래프



(d) "Flamenco2" 비트율-왜곡 그래프

그림 11. IBP 변환 비트율-왜곡 비교 그래프
Fig. 11. RD graph of IBP transcoding

비트율 성능은 61% 향상되었으며, 기존 H.264/AVC의 "IPP" 방법과의 성능보다는 2% 향상된 것을 확인 할 수 있다. GOP 크기가 크기 때문에 "All I" 방법이나 "IPI" 방법 보다는 부호화 효율이 크게 증가했으며, 기존의 "IPP" 방법 보다 실질적인 탐색 영역은 적지만, 보조정보 기반의 예측 움직임 벡터를 중심으로 탐색 영역이 결정되면서, H.264/AVC 기반의 예측 움직임 벡터를 중심으로 결정된 탐색 영역보다 더 먼 영역까지 움직임 추정이 가능해지면서, 제안한 "Pro_IPP" 방법이 기존의 "IPP" 방법 보다 약간의 성능 향상을 보였다.

그림 11은 화면 간 양방향 참조에 의한 변환 방법의 비트율-왜곡 그래프로, 기존 트랜스코딩 방법과 제안한 방법의 성능을 나타낸다. 그래프의 "Pro_IBP"는 제안한 방법의 RD 성능을 나타내며, "All I"는 모든 프레임은 H.264/AVC

의 화면 내 예측 방법으로 부호화를 수행한 결과이다^[11]. "IBP"는 첫 프레임은 H.264/AVC의 화면 내 예측 방법으로 부호화를 수행하고 나머지 프레임들은 H.264/AVC의 화면 간 양방향 예측 방법과 단방향 예측 방법을 반복적으로 부호화를 수행하는 기존의 방법이다^[12]. "IBI"는 기존의 방법으로, GOP 크기가 2로 짝수 프레임은 화면 내 예측 방법으로 부호화를 수행하고, 홀수 프레임은 화면 간 양방향 예측 방법으로 부호화를 수행한 것이다^[13]. 제안하는 "Pro_IBP" 트랜스코딩 방법은 H.264/AVC의 "IBI" 방법보다는 비트율 성능이 62% 향상되었으며, 기존 H.264/AVC의 "IBP" 방법보다는 비트율 성능이 11% 감소되었다. GOP 크기가 증가되면서 기존의 "IBI" 방법보다는 부호화 성능이 크게 증가되었으나, 기존의 "IBP"보다는 성능이 감소되었다. H.264/AVC의 "IBP" 방법에서 높은 선택 비율을 보

표 1. 비트율-왜곡 성능표

Table 1. Rate-Distortion performance table

영상	Ref[11]_IPP		Ref[12]_IPP		Ref[13]_IPI		Ref[11]_IBP		Ref[12]_IBP		Ref[13]_IBI	
	PSNR	RATE	PSNR	RATE	PSNR	RATE	PSNR	RATE	PSNR	RATE	PSNR	RATE
Akko	0.15	-2.88	0.49	-9.09	6.18	-67.44	-0.61	12.51	2.27	-33.94	6.01	-66.05
Ballroom	0.09	-1.95	0.24	-5.21	4.59	-62.31	-0.54	13.23	0.43	-9.17	4.46	-60.92
Exit	0.11	-3.19	0.02	-0.72	4.77	-71.93	-0.31	9.71	0.74	-19.36	4.83	-72.53
Flamenco2	0.08	-1.66	0.11	-2.17	3.29	-46.07	-0.46	10.58	0.70	-14.22	3.47	-48.62
평균	0.11	-2.42	0.21	-4.30	4.71	-61.94	-0.48	11.51	1.03	-19.17	4.69	-62.03

이는 직접예측 모드 혹은 B 스킵이 제안하는 “Pro_IBP” 방법에서는 잘 선택되지 않아서 부호화 효율이 감소되었다. 제안한 방법은 보조정보에서 측정된 양방향 움직임 벡터를 기반으로 예측 움직임 벡터를 결정하게 되고, 이렇게 결정된 예측 움직임 벡터를 기준으로 탐색 영역이 결정되는데, 이 탐색 영역은 직접예측 모드가 선택되는 비율을 감소시킨다. 표 1은 그림 10과 그림 11에서의 나태난 비트율-왜곡 그래프를 비트율-왜곡 성능표로 나타내어 정확한 성능차를 수치로 제시하였다.

표 2는 기존 방법 (기존 화면 내 예측으로 부호화되었던 키 프레임율, 기존 화면 간 예측으로 부호화하는 방법) 과 제안한 방법 (보조정보의 움직임 벡터를 기반으로 화면 간 예측 부호화하는 방법) 의 트랜스코더의 복잡도를 비교한 표이며, 식 (2)를 이용하여 측정되었다.

$$\frac{C_{Conventional} - C_{Proposed}}{C_{Conventional}} \quad (2)$$

표 2. 제안한 방법과 기존 방법들의 복잡도 비교표(%)

Table 2. Complexity comparison of the proposed and conventional methods

영상	QP	복잡도					
		Ref[11]_IPP	Ref[12]_IPP	Ref[13]_IPI	Ref[11]_IBP	Ref[12]_IBP	Ref[13]_IBI
Akko	29	20.11	-3.91	-6.49	26.47	-9.70	-6.83
	33	22.20	-4.96	-6.44	26.54	-9.89	-11.49
	37	23.32	-4.50	-6.40	24.88	-9.99	-16.90
	41	23.37	-3.98	-7.03	20.96	-10.13	-21.64
Ballroom	29	27.66	-4.84	-0.81	36.44	-9.74	-1.81
	33	25.15	-4.25	-2.68	35.81	-9.96	-16.14
	37	18.43	-3.62	-9.16	34.03	-9.91	-28.45
	41	12.30	-5.19	-15.09	30.49	-10.01	-33.26
Exit	29	21.26	-4.67	-4.39	38.16	-10.72	-10.65
	33	19.58	-4.59	-7.84	31.14	-10.11	-16.07
	37	18.06	-4.33	-9.63	23.74	-10.69	-17.13
	41	18.29	-5.13	-10.13	22.40	-9.88	-19.56
Flamenco2	29	26.58	-4.20	0.94	30.34	-10.38	3.31
	33	27.66	-3.69	0.69	22.46	-9.79	-0.69
	37	28.38	-4.80	-0.55	19.29	-10.64	-5.33
	41	28.11	-4.39	0.13	18.42	-10.02	-10.41
평균		22.53	-4.44	-5.30	27.60	-10.10	-13.32

CConventional은 기존 알고리즘의 복잡도를 의미하며, CProposed는 제안한 알고리즘의 복잡도를 의미한다.

표 2의 “Ref[11]_IPP” 방법은 “Ref[11]_IPP (화면 간 단방향 예측 부호화된 기존 방법[11])”과 “Pro_IPP (보조정보 움직임 벡터를 이용해서 화면 간 단방향 예측 부호화된 제안한 방법)”의 복잡도를 비교한 것으로, 제안한 방법이 기존 방법보다 평균 22.53%의 복잡도가 감소하였다. “Ref[12]_IPP” 방법은 “Ref[12]_IPP (보조정보의 움직임 벡터를 이용한 화면 간 단방향 예측 부호화된 기존 방법 [12])”와 “Pro_IPP (보조정보 움직임 벡터를 이용해서 화면 간 단방향 예측 부호화된 제안한 방법)”의 복잡도를 비교한 것으로, 제안한 방법이 기존 방법보다 평균 4.44%의 복잡도가 증가되었다. “Ref[13]_IPI” 방법은 “Ref[13]_IPI (GOP 크기가 2로, 화면 간 단방향 예측 부호화된 기존 방법 [13])”과 “Pro_IPP (보조정보 움직임 벡터를 이용해서 화면 간 단방향 예측 부호화된 제안한 방법)”의 복잡도를 비교한 것으로, 제안한 방법이 기존 방법보다 평균 5.3%의 복잡도가 증가하였다.

“Ref[11]_IBP”방법은 “Ref[11]_IBP (화면 간 단방향과 양방향 예측 부호화된 방법[11])”과 “Pro_IBP (보조정보 움직임 벡터를 이용해서 화면 간 단방향과 양방향 예측 부호화된 제안한 방법)”의 복잡도를 비교한 것으로, 제안한 방법이 기존 방법보다 평균 27.6%의 복잡도가 감소하였다. “Ref[12]_IBP”방법은 “Ref[12]_IBP (보조정보의 움직임 벡터를 고려하여 화면 간 단방향과 양방향 예측 부호화된 방법[12])”과 “Pro_IBP (보조정보 움직임 벡터를 이용해서 화면 간 단방향과 양방향 예측 부호화된 제안한 방법)”의 복잡도를 비교한 것으로, 제안한 방법이 기존 방법보다 평균 10.10%의 복잡도가 증가하였다. “Ref[13]_IBI”방법은 “Ref[13]_IBI (GOP 크기가 2로, 화면 간 양방향 예측 부호화된 방법[13])”과 “Pro_IBP (보조정보 움직임 벡터를 이용해서 화면 간 단방향과 양방향 예측 부호화된 제안한 방법)”의 복잡도를 비교한 것으로, 제안한 방법이 기존 방법보다 평균 13.32%의 복잡도가 증가하였다.

표 1의 비트율-왜곡 성능표와 연계하여, 비트율을 감소하기 위하여 GOP 크기를 확대 및 비트율-왜곡 경쟁을 통하여, 적은 복잡도로 부호화 효율이 증가한 것을 확인할 수

있다. 제안한 방법은 예측 움직임 벡터를 중심으로 하여 넓은 영역에서의 탐색보다는, 블록과의 차가 적을 것으로 예상되는 보조정보 기반의 예측 움직임 벡터를 중심으로 하여, 적은 움직임 탐색을 수행하였기 때문에 기존 방법들의 움직임 추정보다 예측 오차가 적은 장점을 가지고 있다. 또한, 보조정보 기반의 예측 움직임 벡터만이 아니라 기존의 예측 움직임 벡터를 중심으로 한번 더 블록을 탐색함으로써, 움직임 벡터 차분치를 고려한 최적의 부호화 효율을 나타내는 블록을 결정할 수 있었다.

V. 결론

본 논문에서는 보조정보의 움직임 벡터를 이용한 분산 비디오 코딩에서 H.264/AVC로의 변환 방법을 제안한다. 예측 움직임 벡터와 보조정보 기반의 예측 움직임 벡터를 이용하여, 화면 간 단방향 예측 혹은 화면 간 양방향 예측 부호화를 수행함으로써, 제안한 알고리즘은 트랜스코더의 지연을 감소시켰을 뿐만 아니라 부호화 효율도 향상된 것을 확인할 수 있었다. 실험 결과는 기존 변환 방법과 대비하여, 복잡도가 2.82%로 감소하고 비트율 성능은 23.06% 향상되었다. 추후, 제안하는 방법의 복잡도를 유지하면서 기존의 “IBP” 성능과 유사한 성능을 가질 수 있도록 연구를 수행할 것이다. 또한, 실시간 트랜스코딩을 수행하기 위하여, 분산 비디오 복호기의 복잡도를 감소시키고, 복호 정보들을 H.264/AVC의 부호화에서 이용함으로써 더욱 트랜스코딩의 복잡도를 감소시키는 연구를 진행할 것이다.

참고 문헌

- [1] 심혁재, 전병우, “분산 비디오 압축 기술,” 전자공학회지, 제 36권, 4호, 91-105쪽, 2009년 4월
- [2] 박시내, 심동규, 전병우, “조명보상 기반 분산 다시점 비디오 코딩,” 전자공학회논문지, 제 45권, SP편, 제 6호, 17-26쪽, 2008년 11월
- [3] B. Girod, A.M. Aaron, S. Rane, and D. Rebollo-Mondero, “Distributed video coding,” Proc. of the IEEE, vol. 93, pp. 71-83, Jan. 2005
- [4] W.J. Chien, L.J. Karam, and G.P. Abousleman, “Rate-distortion based selective decoding for pixel-domain distributed video coding,” Conf. on ICIP 2008, pp. 1132-1135, Oct. 2008
- [5] J.L. Martinez, C.Holder, G.E. Fernandez, H. Kalva, and F. Quiles,

- “DVC using a half-feedback based approach,” Conf. on Multimedia and Expo, pp. 1125-1128, April 2008
- [6] Q. Linbo, H. Xiaohai, L. Rui, and D. Xiewei, “Application of punctured turbo codes in distributed video coding,” Conf. on ICIG 2007, pp.241-245, Aug. 2007
- [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] 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
- [9] D. Varodayan, A. Aaron, and B. Girod, “Rate-adaptive codes for distributed source coding,” EURASIP Signal Processing Journal, Special Section on Distributed Source Coding, vol. 86, no. 11, pp. 3123-3130, Nov. 2006
- [10] 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), Doc. JVT-G050, Mar. 2003
- [11] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” Trans. on IEEE CSVT, vol. 13, no. 7, pp. 560-576, July, 2003
- [12] E., Pexioto, R.L. Queiroz, and D. Mukherjee, “A Wyner-Ziv video transcoder,” Trans. on CSVT, vol. 20, pp. 189-200, Feb. 2010
- [13] J.L. Martinez, H. Kalva, W.A.C. Fernando, P. Cuenca, and F.J. Quiles, “Efficient WZ-to-H.264 transcoding using motion vector information sharing,” Conf. on ICME 2009, pp. 1394-1397, June, 2009

저 자 소 개



민 경 연

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



유 성 은

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

저 자 소 개



전 병 우

- 1985년 : 서울대학교 전자공학과 졸업(학사)
- 1987년 : 서울대학교 전자공학과 졸업(석사)
- 1992년 : Purdue Univ, School of Elec. 졸업(공학박사)
- 1993년 ~ 1997년 : 삼성전자 신호처리연구소 선임/수석연구원
- 1997년 ~ 현재 : 상균관대학교 정보통신공학부 교수
- 주관심분야 : 멀티미디어 영상압축, 영상인식, 신호처리