

## 적응적 탐색 영역을 이용한 보조 정보 생성의 고속화 방법

박대운<sup>a)</sup>, 심혁재<sup>a)</sup>, 전병우<sup>a)‡</sup>

### Fast Side Information Generation Method using Adaptive Search Range

Daeyun Park<sup>a)</sup>, Hiuk Jae Shim<sup>a)</sup>, and Byeungwoo Jeon<sup>a)‡</sup>

#### 요 약

분산 비디오 압축(DVC, Distributed Video Coding) 기술은 부호화기의 움직임 추정과 같은 복잡한 과정을 복호화기로 이동시킴으로써 경량화 비디오 부호화를 가능하게 하는 새로운 방법으로 주목 받고 있다. 한편, DVC 복호화기에서는 움직임 추정과 보상 과정 뿐만 아니라 채널 복호 과정도 수행해야 하기 때문에 복호화기의 복잡도는 크게 증가하게 된다. 이에 복호화기 복잡도의 가장 많은 비중을 차지하는 채널 복호 과정을 고속화하기 위한 다양한 방법이 제안되고 있다. 하지만 다양한 고속화 방법에 의해 채널 복호 과정이 고속화될수록 채널 복호 이외의 과정들에 대한 복잡도가 부각되는데, 일례로 DVC 복호화기에서 낮은 비율을 차지하던 보조 정보 생성 과정의 복잡도가 상대적으로 증가된다. 따라서 본 논문에서는 보조 정보 생성 과정에 적응적 탐색 영역 방법을 적용하여 DVC 복호 과정을 고속화할 수 있는 알고리즘을 제안한다. 제안된 적응적 탐색 영역을 이용하는 방법을 적용할 경우 고정된 탐색 영역을 적용한 종래의 방법에 비해 평균적으로 보조 정보 생성 시간의 63% 정도가 절감 가능하며, 저하되는 윌 왜곡 성능은 약 0.17% (BDBR) 정도로 상당히 낮다.

#### Abstract

In Distributed Video Coding(DVC), a low complexity encoder can be realized by shifting complex processes of encoder such as motion estimation to decoder. Since not only motion estimation/compensation processes but also channel decoding process needs to be performed at DVC decoder, the complexity of a decoder is significantly increased in consequence. Therefore, various fast channel decoding methods are proposed for the most computationally complex part, which is the channel decoding process in DVC decoding. As the channel decoding process becomes faster using various methods, however, the complexity of the other processes are relatively highlighted. For instance, the complexity of side information generation process in the DVC decoder is relatively increased. In this paper, therefore, a fast method for the DVC decoding is proposed by using adaptive search range method in side information generation process. Experimental results show that the proposed method achieves time saving of about 63% in side information generation process, while its rate distortion performance is degraded only by about 0.17% in BDBR.

Keyword : DVC, WZ, Fast Side Information Generation, Motion Estimation, Adaptive Search Range

a) 성균관대학교 정보통신공학부

School of Information and Communication Engineering, Sungkyunkwan University

‡ 교신저자 : 전병우 (bjeon@skku.edu)

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

· 접수일(2011년12월20일), 수정일(2011년12월23일), 게재확정일(2011년12월23일)

## I. 서론

현재 MPEG-1/2, H.264/AVC와 같은 기존의 비디오 압축 기술은 블록 기반의 움직임 추정 및 보상 과정을 이용하여 부호화 성능을 향상시키지만, 부호화기의 복잡도가 높은 단점이 있다. 따라서 이러한 기존의 비디오 압축 기술은 부호화기의 연산능력이 뛰어난 전통적인 방송 시스템에는 적합하지만, 휴대폰의 카메라, 영상통화, 무선 PC 카메라, 센서 네트워크와 같이 제한된 연산능력의 부호화 시스템에는 적합하지 않다. 이런 이유로 경량화 비디오 부호화를 실현하기 위한 새로운 방법으로 분산 비디오 압축(DVC, Distributed Video Coding) 기술이 주목 받고 있으며, 특히, 주파수 영역에서의 Wyner-Ziv(WZ) 구조를 기반으로 한 연구가 활발히 이루어지고 있다. DVC 부호화기<sup>[1]</sup>는 입력 영상을 키 프레임과 WZ 프레임으로 나누고 키 프레임은 H.26x 등의 인트라 부호화를 사용한다. WZ 프레임은 입력 영상을 주파수 영역으로 변환하고 양자화 한 뒤, LDPC 또는 Turbo 코드와 같은 고성능 채널 코드를 이용하여 부호화한다. WZ 복호화기에서는 전송 받은 키 프레임을 복원하고, 키 프레임간의 시간적 상관성에 기반한 움직임 추정 및 보상 과정을 거쳐 원본과 유사한 보조 정보(Side Information)를 생성한다. 또한 가상의 채널 잡음을 추정하고 채널 복호 과정을 수행하여 보조 정보의 잡음을 제거한다. 따라서 기존의 비디오 압축 기술과 다르게 부호화기의 복잡도를 증가시키는 움직임 추정 및 보상 과정을 부호화기로 이동시킴으로써 경량화 부호화를 실현하였다. 한편 WZ 복호화기에서는 움직임 추정 및 보상 과정뿐만 아니라 채널 복호 과정을 수행해야 하기 때문에 복호화기의 복잡도는 크게 증가하게 된다. 따라서 WZ 복호화기의 복잡도를 감소시키기 위해 다양한 방법이 제안되고 있다. 특히 LDPC 복호 과정을 고속화하기 위한 여러 기술들은 채널 코딩 분야에서도 지속적으로 연구되고 있다. 채널 복호를 고속화하기 위한 기술로는 페리티 요구량을 추정하여 불필요한 복호 과정을 줄이는 방법<sup>[2]</sup>, 경관정 출력을 이용한 고속화 방법(HDA, Hard Decision Aided method)<sup>[3][4]</sup>, 그리고 min-sum 알고리즘<sup>[5]</sup> 등이 있다. 이렇게 다양한 고속화 방법에 의해 채널 복호 과정이 고속화될수록 채널 복호화 이외

의 과정들에 대한 복잡도가 상대적으로 부각되게 되는데, 실제로 이중 WZ 복호화기에서 낮은 비율을 차지하던 보조 정보 생성의 복잡도가 상대적으로 증가되었다.

종래의 보조 정보 생성 과정에서는 이웃하는 키 프레임 간 고정된 탐색 영역으로 움직임 추정이 수행된다<sup>[1]</sup>. 하지만 모든 영상에 필요 이상으로 큰 고정된 탐색 영역으로 움직임 추정이 수행되면, 움직임이 작은 영상에는 보조 정보 생성 과정의 복잡도가 상당히 증가하게 된다. 또한, 탐색 영역의 크기가 증가함에 따라 보조 정보의 화질이 항상 비례하여 증가하지도 않는다<sup>[6]</sup>. [6]에서는 다양한 탐색 영역을 적용하여 그에 따른 보조 정보의 화질 변화를 보여주는 데, 영상에 따라 일정 탐색 영역보다 큰 탐색 영역을 사용하여도 더 이상의 성능향상이 일어나지 않음을 알 수 있다. 이는 일정 수준의 탐색 영역 크기 이후로는 보조 정보의 화질이 수렴되기 때문이다. 따라서 WZ 복호 과정의 고속화를 위해 본 논문에서는 WZ 복호화기의 복잡도를 분석하여 상대적으로 복잡도가 증가하는 보조 정보 생성 과정에 대해 살펴본다. 또한, 보조 정보 생성에 제한 방법을 적용하여 WZ 복호기를 고속화 할 수 있는 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 우선 2장에서는 DVC 복호화기의 복잡도를 분석하고, 3장에서는 보조 정보 생성 과정 및 보조 정보 생성 과정을 이루는 모듈별 복잡도 비율에 대해 살펴본다. 4장에서는 적용적 탐색 영역을 통한 보조 정보 생성의 고속화 알고리즘을 제안하고 동작 원리를 설명한다. 또한, 5장에서는 실험 및 결과를 분석하고 마지막으로 6장에서는 결론을 내린다.

## II. DVC 복호화기의 복잡도 분석

DVC 복호화기는 MPEG-1/2, H.264/AVC와 같은 기존의 비디오 압축 기술과는 다르게 부호화기의 복잡도를 증가시키는 움직임 추정 및 보상 과정이 복호화기에서 수행된다. 또한 복호화기에서 가상의 채널잡음을 추정하여 LDPC 또는 Turbo 코드의 복호 과정을 수행하며, 이에 따라 부호화기의 경량화가 가능하게 되었다. 하지만 복호화기에서 움직임 추정 및 보상 과정과 채널 복호 과정이 수행

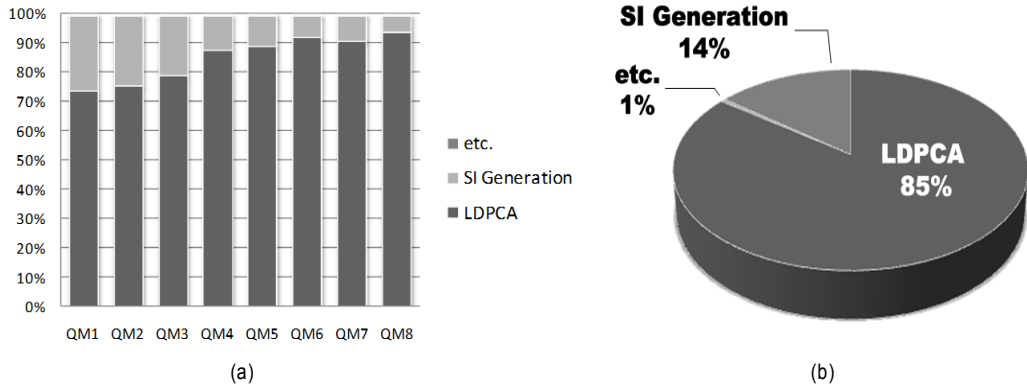


그림 1. 종래의 WZ 복호화기의 복잡도 (a) 양자화 행렬에 따른 복잡도 비율 (b) 평균 복잡도 비율  
 Fig. 1. Conventional WZ decoder complexity (a) Complexity ratio corresponding to QM (b) Average complexity ratio

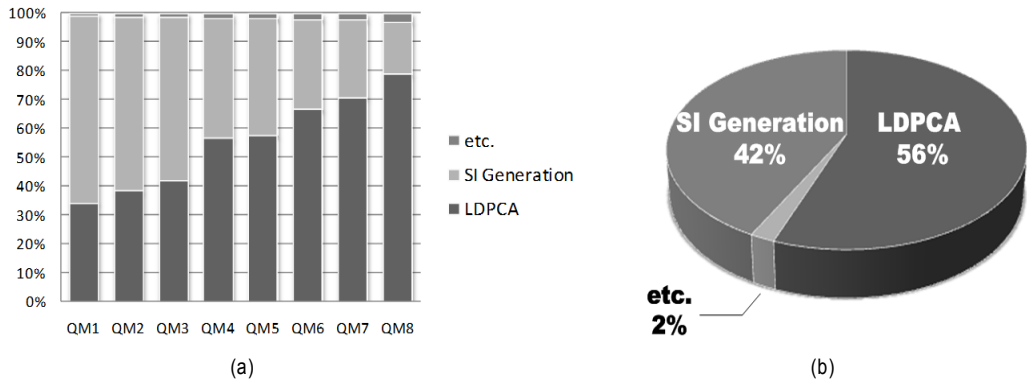


그림 2. HDA와 min-sum을 적용한 WZ 복호화기의 복잡도 (a) 양자화 행렬에 따른 복잡도 비율 (b) 평균 복잡도 비율  
 Fig. 2. WZ decoder complexity applying HDA and min-sum (a) Complexity ratio corresponding to QM (b) Average complexity ratio

되어 WZ 복호화기의 복잡도가 매우 증가하였다. 따라서 WZ 복호화기의 고속화를 위해서는 복잡도 분석을 통하여 높은 복잡도를 요구하는 과정에 대해 살펴볼 필요가 있다.

그림 1(a)는 WZ 복호화기의 복잡도를 양자화 행렬(QM, Quantization Matrix)별로 나타내었으며, QCIF 크기의 Foreman, Coastguard, Hallmonitor, Stefan 영상들의 평균을 나타낸 것이다. 여기서 종래의 WZ 복호화 과정에서는 LDPC 복호 과정이 상당히 높은 복잡도를 차지하고 있다. 그 이유는 LDPC 복호 과정은 확률적인 계산을 통해 얻은 연관성 입력 값을 기반으로 반복적인 belief propagation 과정이 수행(최대 50회)되기 때문이다. 최고 복잡도를 가지는 경우를 생각해보면 총 비트플레인의 개수가 63개(QM8)

이고, 을 적응적인 LDPCA 코드(rate-adaptive LDPCA code)<sup>[7]</sup>를 사용하고 있기 때문에 한 비트플레인 당 최대 65회(QCIF, 1584 길이 LDPC 코드)의 복호 과정이 수행될 수 있다. 따라서 한 WZ 프레임에서 수행되는 LDPC 복호 과정의 최대 횟수는 4,095(63x65)회가 되며, 내부적으로 반복되는 belief propagation 과정의 경우에는 204,750(63x65x50)회의 동일 연산 과정을 수행하게 된다. 따라서 DVC 전체 복잡도의 상당 부분을 차지하는 채널 복호 과정의 고속화를 위한 여러 가지 기술이 제안되고 있는데, 일레로 신드롬 비트 요구량을 추정하여 불필요한 복호 과정을 줄이는 방법과 HDA 방법, 그리고 min-sum 알고리즘 등이 있다.

그림 2(a)는 종래의 WZ 복호화기에 HDA 방법과

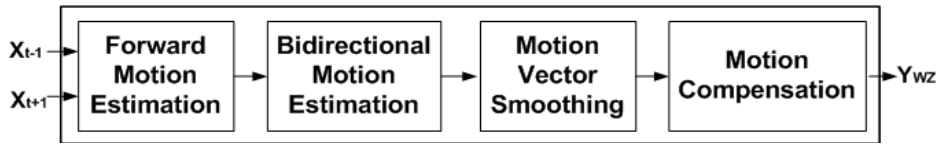


그림 3. 종래의 보조 정보 생성 과정  
Fig. 3. Conventional Side Information Generation

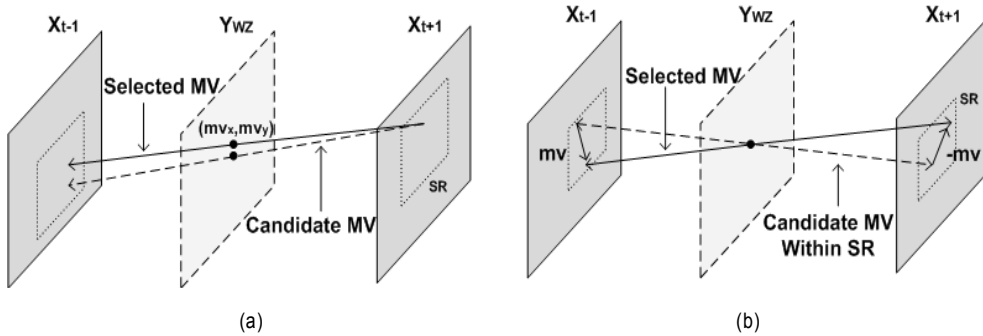


그림 4. (a) 순방향 움직임 추정 (b) 양방향 움직임 추정  
Fig. 4. (a) Forward Motion Estimation (b) Bidirectional Motion Estimation

min-sum 알고리즘의 고속화 방법을 함께 적용한 WZ 복호화기의 복잡도를 보여준다. QM별로 복잡도를 살펴보면, 저 비트율(QM1)에서 고 비트율(QM8)로 증가할수록 LDPC 복호 과정의 복잡도가 증가하므로 보조 정보 생성(SI Generation) 과정의 상대적 복잡도는 감소한다. 여기서 주목할 만한 점은, 종래의 방법(그림 1(a))과 고속화 방법(HDA와 min-sum, 그림 2(a))을 비교했을 때 저 비트율(QM1)에서 보조 정보 생성 과정의 복잡도 차이가 약 40% 정도 난다는 것이다. 특히, QM1, QM2, QM3에서는 보조 정보 생성 과정이 전체 복잡도의 50% 이상을 차지하고 있다. 이것은 저 비트율일수록 보조 정보 생성 과정의 고속화 방법이 더 많은 복잡도 절감 효과를 가진다는 것을 의미한다.

그림 1(b)와 그림 2(b)는 각각 그림 1(a)와 그림 2(a)의 QM별 복잡도의 평균으로 WZ 복호화기의 복잡도를 나타낸 것이다. 그림 1(b)와 그림 2(b)에서도 알 수 있듯이 LDPC 복호 과정은 고속화 방법을 통해 85%에서 56%로 복잡도가 절감되었다. 하지만 상대적으로 보조 정보 생성의 복잡도는 14%에서 42%로 증가하게 되며, LDPC 복호 과정의 고속화 방법이 지속적으로 연구될수록 보조 정보 생성의 복잡도 비율이 상대적으로 증가하게 되는 것을 보

여준다. 따라서 WZ 복호 과정의 고속화를 위해 상대적으로 복잡도가 증가하는 보조 정보 생성 과정의 고속화 방법이 더욱 중요하게 부각될 것이다.

### III. 보조 정보 생성 과정 및 모듈별 비율

본 장에서는 WZ 복호 과정의 고속화에 따라 상대적으로 복잡도가 증가하는 보조 정보의 생성 과정과 이 과정을 이루고 있는 모듈별 복잡도 비율에 대해 살펴본다. 보조 정보를 생성하기 위해서 WZ 복호기에서는 프레임 보간 기술이 사용된다. 프레임 보간 기술에 의해 생성된 보조 정보의 성능은 WZ 부호화의 유효 왜곡 성능을 좌우한다. 이는 동일한 화질대비 적은 비트량이 사용되는 것을 의미하며, WZ 복호기에서 WZ 부호기로 더 적은 양의 패리티 비트를 요구함을 의미한다.

#### 1. 보조 정보 생성 과정

보조 정보 생성 과정은 그림 3과 같이 순방향 움직임 추

정(Forward Motion Estimation), 양방향 움직임 추정(Bidirectional Motion Estimation), 움직임 벡터 평활화(Motion Vector Smoothing), 움직임 보상(Motion Compensation) 과정으로 구성된다<sup>[8]</sup>.

순방향 움직임 추정 과정은 그림 3과 같이 이전 키 프레임( $X_{t-1}$ )과 이후 키 프레임( $X_{t+1}$ )을 입력 받는다. 두 개의 키 프레임을 입력 받은 후 그림 4(a)와 같이 이후 키 프레임을 기준으로 이전 키 프레임 방향으로 움직임을 추정한다. 이때 두 개의 키 프레임간의 움직임은 블록 정합 기준(block matching criteria)을 적용하여 기 정의된 탐색 영역( $\pm 32$ ) 내에서 반화소 정밀도(half-pel precision)로 전 영역 탐색(full search)을 수행한다.

양방향 움직임 추정 과정은 순방향 움직임 추정과정에서 얻은 움직임 벡터를 보조 정보의 각 블록의 위치로 보정하는 과정이다. 하지만 이 과정에서 많은 오차가 발생할 수 있으므로 이러한 오차를 줄이기 위해 움직임 벡터를 보정한다. 움직임 벡터를 보정하는 방법으로는 그림 4(b)와 같이 내삽된 프레임에서 각 블록의 중심을 통과하는 두 개의 키 프레임 간의 선형 궤적을 이용하며, 정제 영역(refinement range) 내에서 움직임 벡터를 보정한다. 이때 주변 움직임 벡터들과의 관계를 통해 정제 영역을 계층적으로 변화시키면서 양방향 움직임 추정 과정이 반복적으로 수행되고, 이를 통해 더 정교한 움직임 벡터로 보정된다<sup>[9]</sup>.

일반적으로 움직임 추정에서 얻은 움직임 벡터는 벡터 필드 내에서 공간적 일관성(spatial coherence)이 적은 경우가 생길 수 있다. 따라서 움직임 벡터 평활화 과정에서는 추정이 잘못된 움직임 벡터의 수를 줄이기 위해 가중 벡터 미디언 필터(Weighted Vector Median Filter)를 적용한다<sup>[10]</sup>.

움직임 보상 과정에서는 이전 키 프레임과 내삽된 프레임간의 시간 간격과, 내삽된 프레임과 이후 키 프레임간의 시간 간격이 같다는 가정 하에 두 개의 키 프레임에 동일한 가중치를 주어 움직임 벡터 보상과정이 이뤄진다<sup>[11]</sup>.

## 2. 보조 정보 생성 과정의 모듈별 복잡도 비율

보조 정보 생성의 고속화를 위하여 보조 정보 생성 과정을 구성하는 네 가지 모듈에 대한 각 모듈별 복잡도 분석이

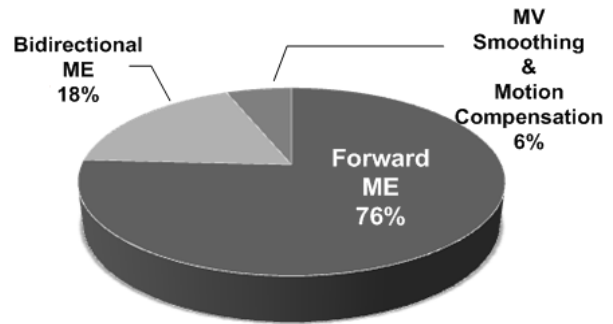


그림 5. 보조 정보 생성 과정의 모듈별 비율 (Foreman, QCIF)  
Fig. 5. Complexity ratio of Side Information Generation module (Foreman, QCIF)

필요하다. 그림 5는 고정된 탐색 영역인  $\pm 32$ 에 대해 QCIF 크기의 Foreman 영상의 양자화 행렬(QM1, QM5, QM7, QM8)의 평균으로 모듈별 복잡도 비율을 나타낸 그림이다.

그림 5와 같이 순방향 및 양방향 움직임 추정과정은 보조 정보 생성 과정의 94% 정도로 높은 복잡도를 차지한다. 이것은 보조 정보 블록들이 키 프레임 간 예측된 움직임 벡터로부터 자신의 위치에 적합한 움직임 벡터를 찾아내기 위해 반복적인 움직임 추정이 수행되기 때문이다. 또한, 순방향 움직임 추정은 양방향 움직임 추정보다 높은 복잡도를 갖는다. 이는 순방향 움직임 추정에서 전 영역을 탐색하기 때문이며, 양방향 움직임 추정에서는 이전 단계에서 찾은 벡터를 정제 영역 내에서 보정하는 과정으로 상대적으로 적은 양의 계산이 요구되기 때문이다. 반면에 움직임 벡터 평활화 과정 및 움직임 보상 과정은 6% 정도의 복잡도를 차지한다. 이것은 이전 단계에서 이미 계산된 움직임 벡터와 SAD(Sum of Absolute Difference) 값이 움직임 벡터 평활화 과정과 움직임 보상 과정으로 전달되어 블록별로 단순 처리되기 때문에 낮은 복잡도를 차지하는 것이다. 본 논문에서는 그림 5와 같이 Foreman 영상에 대한 모듈별 비율만을 보여주고 있는데, 다른 영상(Coastguard, Hallmonitor, Stefan)도 그림 5와 유사한 모듈별 비율을 보인다. 이는 보조 정보 생성 과정을 구성하는 각 모듈별 수행 과정은 고정된 탐색 영역을 적용하기 때문에 영상의 특성과는 상당히 낮은 상관도를 가지기 때문이다.

보조 정보 생성의 움직임 추정 과정은 고정된 탐색 영역( $\pm 32$ )을 사용하여 전 영역 탐색으로 움직임을 추정을 수행하

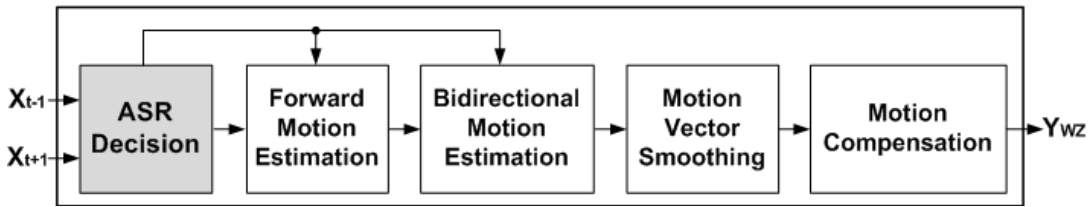


그림 6. 적응적 탐색 영역을 이용한 보조 정보 생성 과정  
 Fig. 6. Side Information Generation using Adaptive Search Range(ASR)

기 때문에 보조 정보 생성 과정 전체의 복잡도가 상당히 증가하게 된다. 이 중 순방향 및 양방향 움직임 추정 과정이 높은 복잡도를 차지한다. 따라서 움직임 추정 과정을 고속 화할 경우 보조 정보 생성 과정 전체의 복잡도를 상당부분 낮출 수 있다.

#### IV. 적응적 탐색 영역을 이용한 제안 방법

DISCOVER 분산 비디오 압축 코덱<sup>[1]</sup>에서는 키 프레임 간의 고정된 탐색 영역을 사용한 움직임 추정을 통해 보조 정보 생성 과정이 이뤄진다. 하지만 영상의 움직임 특성을 고려하지 않은 (즉, 고정된) 탐색 영역으로 움직임 추정을 수행하게 되면, 움직임이 작은 영상에는 보조 정보 생성 과정의 복잡도가 필요 이상으로 증가하게 된다. 또한, 탐색 영역의 크기가 증가함에 따라 보조 정보의 화질이 항상 비례하여 증가하지 않는다. 이는 일정 수준의 탐색 영역 크기 이후로는 보조 정보의 화질이 수렴하기 때문이다. 따라서 움직임 추정을 위한 적절한 크기의 탐색 영역을 적응적으로 결정하게 되면, 보조 정보 생성 과정의 복잡도를 낮추면서 높은 화질의 보조 정보를 생성할 수 있다.

##### 1. 제안 방법

제안 방법은 보조 정보 생성 과정의 고속화를 위해 그림 6과 같이 ASR(Adaptive Search Range) 결정 과정을 거쳐 탐색 영역을 적응적으로 결정하고, 이를 종래의 보조 정보 생성 과정의 순방향 및 양방향 움직임 추정에 적용한다. ASR 결정 과정은 현재 프레임의 움직임 특성을 대표할 수

있는 대표 블록을 설정하는 과정을 수행하게 되는데, 대표 블록은 키 프레임간의 시간적 상관성에 따른 정합 오류(matching error)의 크기에 의해 결정된다. 즉, 현재 프레임 내의 가장 큰 정합 오류 값을 갖는 움직임 추정 블록을 대표 블록으로 설정한다. 이는 일반적으로 정합 오류 값과 움직임 벡터의 크기가 비례하기 때문이다<sup>[12]</sup>. 이것은 TSS(Three Step Search) 나 NTSS(New Three Step Search) 등의 고속 움직임 추정 알고리즘에서 기반하고 있는 가정이다. 일반적으로 이러한 가정의 가장 큰 문제점은 국소 최저치(local minima)에 빠질 경우 최적의(optimal) 움직임 벡터를 추정할 수 없다는 것이지만, 제안 방법은 탐색 지점(search point)을 줄이는 알고리즘이 아니라 기본적으로 전 영역 탐색에 기반하고 있기 때문에 TSS 등의 알고리즘이 가지는 문제점을 가지지 않는다. 또한 본 논문에서는 정합 오류의 크기를 측정하기 위해 SAD를 사용하는데, 이는 연산이 간단하고 이웃하는 키 프레임 사이에 내삽된 프레임의 움직임 특성을 잘 반영하기 때문이다. 따라서 SAD가 가장 큰 블록(대표 블록)을 이용하여 탐색 영역을 결정하게 되면, 현재 프레임의 움직임 특성을 대부분 반영하는 적응적 탐색 영역을 결정할 수 있다.

그림 7은 ASR 결정 과정을 나타낸 것으로 대표 블록을 설정하기 위해 초기 움직임 벡터 탐색(Initial MV Search) 과정을 수행한다. 초기 움직임 벡터 탐색 과정은 이웃하는 키 프레임 간의 작은 탐색 영역(±4)을 사용한 순방향 움직임 추정을 통해 대표 블록을 설정하는 과정이다. 종래의 보조 정보 생성은 현재 프레임의 모든 움직임 추정 블록에 대해 고정된 탐색 영역으로 움직임 추정을 수행하였다. 따라서 움직임이 작은 영상에는 고정된 탐색 영역으로 인해 보조 정보 생성에 따른 계산량이 필요이상으로 많아진다.

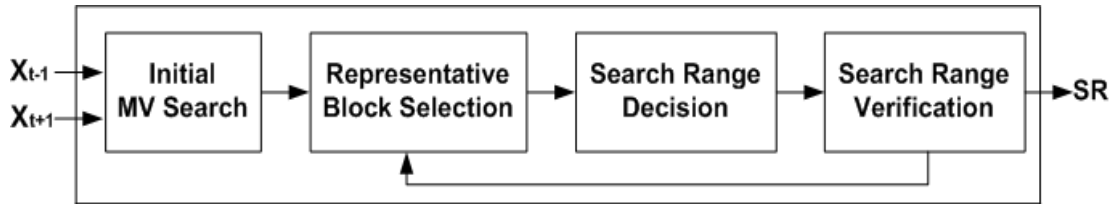


그림 7. 적응적 탐색 영역의 결정 과정  
Fig. 7. Adaptive Search Range(ASR) Decision Process

하지만 현재 프레임의 움직임 특성을 대표하는 블록을 설정하여 영상의 특성에 맞는 탐색 영역을 적응적으로 선정하여, 보조 정보 생성에 따른 계산량을 줄일 수 있다.

대표 블록 선택(Representative Block Selection) 과정에서는 초기 움직임 벡터 탐색 과정에서 얻은 모든 움직임 추정 블록에 대한 SAD값들을 크기 순서로 분류한다. 이렇게 분류된 SAD값 중 가장 큰 SAD값을 갖는 움직임 추정 블록을 현재 프레임의 대표 블록으로 설정한다. 여기서 가장 큰 SAD값을 대표 블록으로 설정하는 것은 SAD값이 클 수록 키 프레임 간의 시간적 상관성이 적어 그에 따라 움직임 벡터가 커지기 때문이다. 이는 대표 블록의 움직임 벡터가 다른 움직임 추정 블록의 움직임 벡터보다 대체적으로 크다는 것을 의미한다. 따라서 대표 블록의 움직임 벡터만큼의 크기로 탐색 영역을 선택하면 현재 프레임의 움직임을 대부분 반영하게 된다.

이렇게 대표 블록이 결정되면 다음으로 탐색 영역 결정(Search Range Decision) 과정이 수행된다. 탐색 영역 결정 과정은 대표 블록의 움직임 벡터 성분의 최대값( $MV_{max}$ )을 구하고, 이를 통해 적응적 탐색 영역을 결정하는 과정이다. 움직임 벡터(MV)는 x성분( $MV_x$ )과 y성분( $MV_y$ )으로 구성되는데,  $MV_{max}$ 는 움직임 벡터의 각 성분의 크기 ( $|MV_x|, |MV_y|$ )가 식 (1)에 적용되어 구해진다.

$$MV_{max} = \max\{|MV_x|, |MV_y|\} \quad (1)$$

여기서,  $\max\{A,B\}$ 는 A와 B를 비교하여 더 큰 값을 선택하는 연산을 의미한다.

움직임 벡터 성분의 최대값( $MV_{max}$ )을 구하기 위해서는 대표 블록에 대해 탐색 영역을  $\pm 32$ 로 확장 적용하여 순방

향 움직임 추정을 수행하며, 이렇게 얻어진  $MV_{max}$ 로 식 (2)의 조건에 따라 탐색 영역(SR: Search Range)의 후보를 결정한다. 예를 들어,  $MV_{max}$ 의 값이 20으로 구해지면 탐색 영역의 후보는  $\pm 24$ 로 결정되는 것이다. 하지만 한가지 예외 사항이 있는데, 초기 탐색 영역( $\pm 4$ )으로 구한  $MV_{max}$ 가 4보다 작을 경우에는 탐색 영역을  $\pm 32$ 로 확장 적용하지 않고  $\pm 4$ 를 적응적 탐색 영역으로 결정한다(즉, ASR 과정을 종료). 그리고 이를 순방향 및 양방향 움직임 추정에 적용하여 보조 정보 생성 과정을 수행하는 것이다. 이는 작은 탐색 영역( $\pm 4$ )으로 얻은 대표 블록이 현재 프레임의 움직임 특성을 잘 반영한다고 판단할 수 있기 때문이다. 하지만  $\pm 4$ 의 탐색 영역으로 구한  $MV_{max}$ 가 4와 같을 경우에는, 탐색 영역이 대표 블록의 움직임 벡터를 반영할 만큼 크지 않다고 판단할 수 있다. 따라서 이런 경우에는 예외 사항이 적용되지 않고, 선택된 대표 블록에 대해 탐색 영역의 크기를  $\pm 32$ 까지 늘려  $MV_{max}$ 를 찾고 식 (2)의 조건을 만족하는 탐색 영역의 후보를 얻는다.

$$SR = \begin{cases} \pm 4, & 0 \leq MV_{max} < 4 \\ \pm 8, & 4 \leq MV_{max} \leq 8 \\ \pm 16, & 8 < MV_{max} \leq 16 \\ \pm 24, & 16 < MV_{max} \leq 24 \\ \pm 32, & 24 < MV_{max} \leq 32 \end{cases} \quad (2)$$

탐색 영역의 후보를 식 (2)와 같이 선택한 이유는 다음과 같다.  $\pm 4$ 의 경우, 제안 방법을 적용하기 위한 초기 탐색 영역으로 정합 오류의 크기에 따라 현재 프레임의 특성을 대표할 수 있는 블록을 선택하기 위해 사용된다. 또한 나머지 탐색 영역의 후보들은  $\pm 32$ 까지의 탐색 영역 내에서 균등한 크기로 정하였다.

표 1. 적응적 탐색 영역의 결정 방법  
Table 1. Adaptive Search Range Decision Method

Initial MV Search	step1	초기 탐색 영역을 $\pm 4$ 로 하여 현재 프레임( $X_{t-1}$ ) 내의 모든 움직임 추정 블록에 대해 순방향 움직임 추정을 수행한다.
Representative Block Selection	step2	step1에서 얻은 모든 움직임 추정 블록에 해당하는 SAD값을 큰 순서대로 분류하고, 이 과정에서 가장 큰 SAD를 갖는 움직임 추정 블록을 적응적 탐색 영역을 얻기 위한 현재 프레임의 대표 블록으로 선택한다.
Search Range Decision	step3	선택된 대표 블록에 대해 $\pm 32$ 의 탐색 영역으로 순방향 움직임을 추정하여 식 (1)에 의해 $MV_{max}$ 를 얻고, 식 (2)의 조건에 따라 탐색 영역의 후보를 얻는다. ※예외 사항 : $MV_{max}$ 의 값이 4보다 작을 경우 적응적 탐색영역을 $\pm 4$ 로 결정하고 ASR 과정을 종료한다.
	step4	대표 블록이 적절하게 설정되었는지 검증하기 위해, 대표 블록의 SAD값 다음으로 큰 SAD를 갖는 움직임 추정 블록을 새로운 대표 블록으로 설정하여 step3을 반복한다.
Search Range Verification	step5	이전의 탐색 영역 후보와 새로운 탐색 영역 후보를 비교한다. 이전의 탐색 영역 후보가 새로운 탐색 영역 후보보다 크거나 같다면 이전의 탐색 영역 후보를 적응적 탐색 영역으로 결정하고 ASR 과정을 종료한다.
	step6	만약 새로운 탐색 영역 후보가 이전의 탐색 영역 후보보다 크다면, step5가 만족할 때까지 step4~5를 반복하여 적응적 탐색 영역을 결정하고 ASR 과정을 종료한다.

적응적 탐색 영역을 얻기 위한 마지막 단계로는 탐색 영역 검증(Search Range Verification) 과정이 수행된다. 탐색 영역 검증 과정은 대표 블록 선택 과정에서 SAD값에 따라 분류된 움직임 추정 블록이 대표 블록으로 알맞게 설정되었는지 검증하는 과정이다. 즉, 충분히 큰 움직임을 갖는 블록이 대표 블록으로 선택되지 않았다면, 탐색 영역 결정 과정을 통해 얻은 탐색 영역(SR)이라 할지라도 현재 프레임의 움직임 특성을 충분히 반영하지 못할 가능성이 있다. 이는 대표 블록 선택 과정에서  $\pm 4$ 의 탐색 영역을 이용하여 대표 블록을 결정하였기 때문이며, 탐색 영역이  $\pm 32$ 로 늘어났을 때는 다른 움직임 추정 블록이 현재 프레임의 특성을 더 잘 반영할 수 있기 때문이다. 따라서 선택된 대표 블록이 현재 프레임을 대표하는지 검증하기 위해, 대표 블록 선택 과정에서 분류된 SAD값 중 다음으로 큰 SAD값을 갖는 움직임 추정 블록을 새로운 대표 블록으로 선택하고, 이에 대해 탐색 영역 결정 과정(Search Range Decision)을 반복해 새로운 탐색 영역의 후보를 얻는다. 이렇게 얻은 새로운 탐색 영역의 후보와 이전의 대표 블록을 통해 얻은 탐색 영역의 후보를 비교하게 되는데, 만약 이전의 대표 블록을 통해 얻은 탐색 영역의 후보가 더 크거나 같다면 충분한 크기의 탐색 영역이 선택된 것으로 판단한다. 따라서 이를 최종적인 적응적 탐색 영역으로 결정하여 보조 정보 생성 과정을 수행한다. 하지만 탐색 영역 검증 과정을 통해 얻은

새로운 탐색 영역의 후보가 더 크다면, 새로운 대표 블록이 현재 프레임의 움직임 특성을 잘 반영하는지 다시 검증 과정을 수행한다. 결국 탐색 영역 검증 과정은 현재 프레임을 대표하는 블록이 적절하게 선택될 때까지 반복되어 최종적으로 현재 프레임의 움직임을 잘 반영하는 적응적 탐색 영역을 결정하게 된다.

ASR 결정 과정을 통해 최종적으로 얻은 적응적 탐색 영역은 보조 정보 생성의 순방향 움직임 추정과 양방향 움직임 추정에 각각 적용된다. 이때 순방향 움직임 추정에는 ASR 결정 과정에서 구한 적응적 탐색 영역(최종 탐색 영역 후보인 SR)이 적용되며, 양방향 움직임 추정에는 적응적 탐색 영역의 절반(SR/2)을 적용하여 움직임 벡터를 보정하는 목적의 정제 영역으로 사용된다. 표 1은 ASR 결정 과정을 단계별로 정리한 것이다.

본 논문에서 제안하는 적응적 탐색 영역을 결정하는 방법에 있어, 그 전체 과정 자체를(그림 7) 병렬화 하는 것은 가능하지 않다. 하지만 개개 블록의 경우에는 병렬 처리가 가능하며, 특히 높은 복잡도를 요구하는 초기 움직임 벡터 탐색(Initial MV Search) 블록과 탐색 영역 결정(Search Range Decision) 블록의 경우에는 병렬 처리가 용이하기 때문에 제안 방법을 더욱 고속화하는 것이 가능할 것이다. 이는 기존 비디오 압축에서 이용하는 pmv(predicted motion vector)를 이용하지 않기 때문에 움직임 추정 시 이웃



표 2. 실험 조건  
Table 2. Experimental Conditions

Sequence	Foreman	Coastguard	Hallmonitor	Stefan
Resolution	QCIF(176x144)			
Frames	150			
ME block size	16x16			
GOP	2			
Quantization Matrix <sup>[13]</sup>	1, 5, 7, 8			
Quantization Parameter (intra frame)	40,34,29,25	38,33,30,26	37,33,29,25	40,34,29,25

블록 간의 의존성이 없기 때문이다. 따라서 초기 움직임 벡터 탐색 과정이 모든 움직임 추정 블록에 대해 병렬적으로 수행될 수 있으며, 또한 탐색 영역 결정 및 탐색 영역 검증에서 대표 블록에 대해 순차적으로 진행되었던 과정을 여러 블록에 대해 병렬 처리하게 되면 제안 방법을 더욱 고속화 할 수 있다.

## V. 실험 결과 및 분석

### 1. 실험 조건

본 논문에서는 분산 비디오 압축 복호화 과정 중 보조 정보 생성 과정의 고속화 방법에 대한 성능 분석을 위해 Foreman, Coastguard, Hallmonitor, Stefan 등 총 4개의 영상에 대하여 실험하였다. 실험에 사용된 영상의 크기는 QCIF(176x144)이고, 총 150프레임에 프레임 율은 15Hz를 사용하였다. 성능 비교의 기준이 되는 실험(Anchor)<sup>[1]</sup>의 조건으로 움직임 추정의 블록 크기는 16x16, 순방향 움직임 추정 시 탐색 영역은 ±32, 양방향 움직임 추정의 탐색영역은 ±16으로 하였다. 또한, GOP(Group of Picture)의 크기는 2로 고정하였다. 자세한 실험 조건은 표 2에서 확인할 수 있다.

### 2. 제안 방법에 따른 복잡도 및 율 왜곡 성능 분석

표 3과 표 4는 적응적 탐색 영역을 이용한 제안 방법을 적용 했을 때의 WZ 프레임에 대한 결과이다. 율 왜곡 성

능은 BDPSNR(Bjontegaard Delta PSNR) 및 BDBR (Bjontegaard Delta Bitrate)<sup>[14]</sup>로 나타내었고, 보조 정보 생성 과정에 대한 시간 절감율(TSIG, Time Saving of Side Information Generation)은 식 (3)과 같이 계산하였다.

$$TSIG = \frac{T_{Prev} - T_{Proposed}}{T_{Prev}} \times 100 [\%] \quad (3)$$

여기서 TSIG는 보조 정보 생성의 시간 절감율이고, T<sub>Prev</sub>는 종래의 보조 정보 생성 과정<sup>[1]</sup>의 시간이며, T<sub>Proposed</sub>는 제안 방법을 사용한 경우 보조 정보 생성 과정의 시간을 나타낸다.

표 3은 제안 방법에 의해 영상마다 움직임 벡터의 크기에 따른 적응적 탐색 영역의 선택 비율을 나타낸다. 탐색 영역이 ±32보다 작은 경우(±4~±24)의 적응적 탐색 영역의 선택 비율을 보면, Foreman은 74.32%, Coastguard는 94.59%, Hallmonitor는 97.97%, Stefan의 경우 70.95%가 된다. Coastguard와 Hallmonitor 영상은 Foreman과 Stefan 영상과는 달리 ±32보다 작은 탐색 영역이 더 많이 선택되었다. 이는 움직임이 적은 영상에서는 ±32보다 작은 크기의 탐색 영역이 적응적으로 선택되기 때문이다. 이외는 다르게 Foreman과 Stefan의 경우 ±32인 탐색 영역이 상당 부분 선택된다. 이것은 움직임이 많은 영상에서는 보조 정보 생성 시, 키 프레임간의 시간적 상관성이 적어 블록별 SAD가 커짐에 따라 움직임 벡터가 커져서 보다 큰 탐색 영역이 선택되기 때문이다. 또한, 전체적으로 모든 영상에 대해 ±32보다 작은 크기의 탐색 영역이 선택되는 비율은 평균 84.47%이며 특히, 탐색 영역이 ±4로 선택되는 경우가 평균

표 3. 적응적 탐색 영역 선택 비율 [%]  
Table 3. Ratio of selected adaptive search range [%]

Sequence	±4	±8	±16	±24	±32
Foreman	27.70	4.05	25.00	17.57	25.68
Coastguard	21.62	37.36	25.00	10.61	5.41
Hallmonitor	89.19	4.73	1.35	2.70	2.03
Stefan	14.87	10.81	29.05	16.22	29.05
Average	38.35	14.24	20.10	11.78	15.54
Cumulated average	38.35	52.58	72.68	84.46	100.00

표 4. 제안 방법에 따른 올 왜곡 성능 및 시간 절감율  
Table 4. RD performance and time saving of the proposed method

Sequence	BDPSNR[dB]	BDBR[%]	TS <sub>sig</sub> [%]
Foreman	0.01	-0.18	46.89
Coastguard	0.00	-0.08	72.69
Hallmonitor	0.00	0.01	80.74
Stefan	-0.06	0.93	48.82
Average	-0.0125	0.17	62.29

38.35%로 가장 많이 선택되는 것을 알 수 있다.

표 4는 제안 방법에 따른 올 왜곡 성능 및 보조 정보 생성 시간 절감율을 나타낸다. ±32로 고정된 탐색 영역을 사용한 실험과 비교하여 BDBR은 평균적으로 약 0.17% 증가하지만, 보조 정보 생성 과정이 가지는 복잡도의 62.29%를 절감할 수 있다. 특히, Hallmonitor 영상은 적응적 탐색 영역이 ±4로 선택된 경우가 89.19%인데 매우 적은 성능 저하에도 불구하고 80.74%의 시간 절감이 되었다. 움직임이 적은 영상일수록 작은 크기의 탐색 영역이 선택되고, 그에 따라 많은 시간 절감의 이득을 볼 수 있다. 표 4를 보면 Foreman과 Coastguard 영상의 경우에는 많은 시간 절감에도 오히려 성능이 약간 증가하는 것을 관찰할 수 있다. 이는 WZ 복호화기의 경우, 보조 정보 생성과 대응되는 원본 영상에 대한 정보가 없기 때문에 이전 키 프레임과 이후 키 프레임만으로 움직임을 추정하고 있기 때문이며, 이러한 움직임 추정 결과로 생기는 영상과 원본 영상이 항상 유사하다고 보장할 수 없기 때문이다. 따라서 보조 정보 생성의 경우 탐색 영역이 클수록 항상 좋은 화질의 보조 정보가 생성되는 것은 아니며, 오히려 적정 크기의 탐색 영역이 주

어지는 경우 보조 정보의 화질이 향상될 수 있다. 따라서 Foreman과 Coastguard 영상은 이러한 경우에 해당하여 성능이 증가하는 것이다.

결론적으로 보조 정보 생성 시, 필요 이상의 크기와 고정된 탐색 영역으로 움직임을 추정하는 것보다 영상의 특성에 따라 적응적 탐색 영역으로 움직임 추정을 하는 것이 시간 절감이 크며 그에 따른 성능 저하도 상당히 낮다.

## VI. 결 론

본 논문에서는 분산 비디오 압축 복호화기의 복잡도 분석을 통해 채널 복호 과정 및 보조 정보 생성 과정이 전체 복호화 과정의 대부분을 차지하고 있음을 확인하였다. 기존 연구에 의하면 채널 복호 과정의 경우, HDA방법과 min-sum알고리즘을 통해 85%에서 56%로 복잡도를 절감할 수 있지만, 상대적으로 보조 정보 생성의 복잡도가 14%에서 42%로 증가하기 때문에 WZ 복호 과정에서 보조 정보 생성의 복잡도가 부각되었으며, 채널 복호화의 고속화

연구가 진행될수록 이러한 현상은 가속화될 것이다. 따라서 본 논문에서는 보조 정보 생성 과정에서 큰 비중을 차지하는 순방향 및 양방향 움직임 추정 과정에 적용 가능한 적응적 탐색 영역의 결정 방법을 제안하였다. 제안 방법은 초기 탐색 영역을 적용하여 선택된 대표 블록을 통하여 탐색 영역을 적응적으로 결정하는 것이다. 적응적 탐색 영역을 통한 보조 정보 생성 방법은 고정 탐색 영역( $\pm 32$ )을 사용한 방법에 비해 평균 약 63% 정도의 시간 절감을 하는 반면, BDBR이 평균 0.17% 정도로 성능 저하가 크지 않았다. 실험 결과를 통해 보조 정보 생성 시, 움직임 추정 과정에 적응적 탐색 영역을 적용함으로써 고속화된 보조 정보 생성 과정을 실현할 수 있으며, 이를 통해 WZ 복호화기를 고속화할 수 있었다. 향후 적응적 탐색 영역 내에서 신뢰할 만한 성능의 보조 정보를 생성하는 시점에서 적응적으로 초기 종결을 하는 방법을 통해 더욱 고속화된 WZ 복호화기의 구현에 대한 연구가 필요하다.

### 참 고 문 헌

[1] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, M. Ouaret, "The DISCOVER codec: Architecture, Techniques and Evaluation," Picture Coding Symposium, November, 2007.

[2] 심혁재, 오양근, 전병우, "패리티 요구량 추정을 이용한 적응적 경관정 출력 기반 고속 분산 비디오 복호화 기술", 방송공학회논문지 제16권 제4호, 2011년 7월.

[3] R.Y.Shao, S.Lin and M.P.C. Fossorier, "Two simple stopping criteria for turbo decoding," IEEE Trans. Comm, vol. 47, pp. 1117-1120, August 1999.

[4] 오양근, 심혁재, 전병우, "적응적 경관정 출력을 이용한 고속 분산 비디오 복호화 기술", 전자공학회 논문지 제 47권 SP 편 제 6호, 2010년 11월.

[5] William E. Ryan, Shu Lin, Channel Codes Classical and Modern, Cambridge University Press, The Edinburgh Building, Cambridge CB2 8RU, UK, pp.692, 2009.

[6] X. HoangVan, J. Park, B. Jeon, "A flexible side information generation scheme using adaptive search range and overlapped block motion compensation", Proc. The 5th International Conference on Ubiquitous Information Management and Communication, Article No. 46, SKKU, Suwon, Korea, February 21-23, 2011.

[7] D. Varodayan, A. Aaron and B. Girod, "Rate- adaptive codes for distributed source coding," EURASIP Signal Processing Journal, vol. 86, no. 11, pp. 3123-3130, November 2006.

[8] C. Brites, J. Ascenso, F. Pereira, "Evaluating a feedback channel based transform domain Wyner-Ziv video codec," Image Communication vol. 23, no. 4, pp.269-297, April 2008.

[9] J. Ascenso, C. Brites, F. Pereira, "Content adaptive Wyner-Ziv video coding driven by motion activity", International Conference on Image Processing, Atlanta, USA, October 2006.

[10] L. Alparone, M. Barni, F. Bartolini and V. Cappellini, "Adaptively Weighted Vector-Median Filters for Motion Fields Smoothing," IEEE ICASSP, Georgia, USA, pp. 2267-2270, May 1996.

[11] J. Ascenso, C. Brites, F. Pereira, "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding", Computer and Information Science, Issue.Dvc, pp.1-6, June 2005.

[12] Reoxiang Li, Bing Zeng, Liou, M.L., "A new three-step search algorithm for block motion estimation", Circuits and Systems for Video Technology, IEEE Transactions on, vol. 4, no. 4, pp. 438-442, August 1994.

[13] C. Brites, J. Ascenso and F. Pereira, "Improving transform domain Wyner-Ziv coding performance," IEEE International Conf. on Acoustic Speech and Signal Processing, pp. 525-528, Toulouse, France, May 2006.

[14] G. Bjontegaard, "Calculation of Average PSNR Differences Between RD-curves," document VCEG-M33, ITU-T Video Coding Experts Group (VCEG) Meeting, Austin, TX, April 2-4, 2001.

### 저 자 소 개



박 대 운

- 2010년 : 성균관대학교 전자전기공학과 졸업 (학사)
- 2011년 ~ 현재 : 성균관대학교 휴대폰학과 석사과정
- 주관심분야 : 멀티미디어 영상압축, 신호처리

---

저 자 소 개

---



심 혁 재

- 2000년 : 성균관대학교 전자공학과 졸업 (학사)
- 2002년 : 성균관대학교 정보통신공학부 졸업 (석사)
- 2002년 ~ 현재 : 성균관대학교 정보통신공학부 박사과정
- 주관심분야 : 멀티미디어 영상압축, 신호처리



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 수석연구원
- 1997년 ~ 현재 : 성균관대학교 정보통신공학부 교수
- 주관심분야 : 멀티미디어 영상압축, 영상인식, 신호처리