

일반논문-11-16-1-14

병렬 LDPCA 채널코드 부호화 방법을 사용한 고속 분산비디오부호화

박종빈^{a)}, 전병우^{a)‡}

Fast Distributed Video Coding using Parallel LDPCA Encoding

Jongbin Park^{a)} and Byeungwoo Jeon^{a)‡}

요약

본 논문에서는 고속, 저전력 비디오 부호화에 적합한 변환영역 Wyner-Ziv 분산비디오부호화기를 더욱 고속화하기 위한 병렬처리 방법을 제안한다. 기존의 변환영역 Wyner-Ziv 분산비디오부호화 방법은 양자화 된 변환계수를 비트플레인 단위로 분해한 후 비트플레인별로 순차적으로 LDPCA 채널코드로 부호화함에 따라 전체 부호화 연산량에서 LDPCA 부호화가 평균적으로 60% 정도 차지하였고, 이러한 복잡도는 고비트율로 부호화 할수록 더욱 증가하였다. 본 논문에서는 이런 분산비디오부호화 방법의 복잡도 문제를 개선하기 위해 여러 개의 비트플레인들을 하나의 메시지묶음으로 묶어서 한 번의 연산으로 여러 개의 데이터를 동시에 고속 LDPCA 채널코드 부호화하는 병렬화 방법을 제안한다. 이를 통해 기존의 순차적 방법에 비해 저비트율에서는 8배, 고비트율에서는 55배까지 LDPCA 채널코드 부호화 속도를 향상시켰다. 결과적으로 전체 변환영역 Wyner-Ziv 분산비디오부호화에서 LDPCA 채널코드 부호화의 상대적인 복잡도 비율을 평균 9%까지 낮출 수 있었으며, Wyner-Ziv 영상의 부호화 속도도 QCIF 크기 영상을 2.5GHz 속도의 CPU를 가진 PC환경에서 GOP 길이가 64인 경우 초당 700 ~ 2,300장을 부호화 할 수 있음을 확인했다. 제안 방법은 LDPCA를 사용하는 화소영역 Wyner-Ziv 분산비디오부호화에도 적용 가능하여 고속의 부호화가 요구되는 다양한 응용에 활용이 기대된다.

Abstract

In this paper, we propose a parallel LDPCA encoding method for fast transform-domain Wyner-Ziv video encoding which is suitable in an ultra fast and low power video encoding. The conventional transform-domain Wyner-Ziv video encoding performs LDPCA channel coding of quantized transform coefficients in bitplane-serial fashion, which takes about 60% of total encoding time, and this computational complexity becomes severer as the bitrate increases. The proposed method binds several bitplanes into one packed message and carries out the LDPCA encoding in parallel. The proposed LDPCA encoding method improves the encoding speed by 8 ~ 55 times. In the experiment, the proposed Wyner-Ziv encoder can encode 700 ~ 2,300 QCIF size frames per second with GOP=64. The method can be applied to the pixel-domain Wyner-Ziv encoder using LDPCA, and has a wide scope of application.

Keyword : Distributed Video Coding, Parallel LDPCA, Transform Domain Wyner-Ziv, SIMD

a) 성균관대학교 정보통신공학부
Sungkyunkwan University, School of Information and Communication Engineering

‡ 교신저자 : 전병우 (bjeon@skku.edu)

※ 본 연구는 지식경제부, 방송통신위원회 및 한국산업기술평가관리원의 산업원천기술개발사업(정보통신)(K1002142, 차세대 모바일 영상서비스를 위한 초경량 비디오 부호화 원천기술개발)과 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(NIPA-2010-(C1090-1021-0008))(NTIS-2010-(1415109527))
· 접수일(2010년12월14일), 수정일(2011년1월4일), 게재확정일(2011년1월4일)

1. 서론

분산소스부호화(DSC, Distributed source coding)는 서로 독립적으로 부호화를 수행하더라도 신호사이의 상관성을 이용한 결합 복호화를 수행한다면, 각 소스를 결합하여 부/복호화하는 것과 같은 부호화 성능을 달성 할 수 있다는

이론에 기초한다^[1,2]. 이러한 개념은 비디오 신호의 압축에 적용되어 분산비디오부호화(DVC, Distributed video coding) 라는 주제로 다양한 연구가 진행 중이다^[3-8]. 통상적으로 H.264/AVC^[9]와 같은 전통적인 비디오부호화 방법은 부호화기에서 움직임예측 및 보상과 같은 예측부호화를 수행함으로써 부호화 성능을 향상시키는 결합 부호화 방법(Joint encoding)이다. 그러나, 예측부호화에는 많은 연산이 필요하므로 제한된 전력을 사용해야 하거나 고속으로 부호화가 필요한 응용에서는 큰 제약요소가 될 수 있다. 따라서, 이러한 응용들에 대해서는 예측부호화가 불필요한 분산비디오부호화와 같은 저복잡도 부호화기가 매우 적합한 방법이라 할 수 있다. 그럼에도 불구하고 분산비디오부호화의 효과적인 구현과 관련된 연구는 아직 상대적으로 미흡하다.

기존의 변환영역 Wyner-Ziv(TDWZ, Transform domain Wyner-Ziv) 분산비디오부호화기는 입력 영상을 DCT와 같은 변환을 사용하여 변환계수로 전환한 후, 변환 계수별로 적절한 양자화를 수행하여 전송해야할 데이터량을 조절한 후 양자화된 정보를 비트플레인 단위로 분해 하고 이를 터보코드(Turbo code)나 저밀도 패리티 체크(LDPC, Low density parity check) 코드 등을 사용하여 순차적으로 채널코드 부호화 한다^[3-8]. 따라서 전송해야 할 비트량이 증가하면 채널코드 부호화를 수행하는데 소요되는 연산량 역시 크게 증가하였다. 본 논문을 위한 사전과정으로, 정수 DCT 변환(Integer discrete cosine transform)과 LDPCA(Rate adaptive LDPC) 채널코드 부호화 방법^[10,11]을 사용한 TDWZ 부호화기를 구현하여 실험한 결과 전체 부호화기 연산량에서 LDPCA 채널코드 부호화에 소요되는 연산량의 비율은 평균 60%에 달하여(저비트율에서는 약 54%, 그리고 고비트율에서는 약 67%까지 증가함을 관찰함) 이러한 문제점에 대한 지적이 올바르게 실험적으로 확인하였다. 이와 같은 문제를 해결하기 위해 본 논문에서는 여러 개의 비트플레인들을 하나의 메시지묶음(Packed message)으로 구성한 후, 동시에 병렬로 LDPCA 채널코드 부호화를 수행하는 방법을 제안한다. 이는 한 번의 연산으로 여러 개의 데이터를 동시에 처리하는 단일 명령 복수 데이터 처리(SIMD, Single instruction, multiple data)에 의한 고속화 방

법^[12]이지만, 본 논문에서 제안하는 LDPCA 병렬화 방법은 특정 하드웨어 아키텍처에 의존하지 않고 일반적인 컴퓨터 환경에서 LDPCA 부호화를 병렬로 수행할 수 있다는 점이 중요한 특징이다.

제안방법은 기존의 순차적 처리 방법에 비해 저비트율에서는 8배, 고비트율에서는 55배까지 채널코드 부호화 속도를 향상시켰다. 따라서 전체 부호화에서 LDPCA의 상대적인 복잡도 비율은 저비트율에서는 14%, 고비트율에서는 6%정도까지 낮아지게 되었다. 또한, Wyner-Ziv 영상의 부호화 속도도 QCIF 크기 영상을 2.5GHz 속도의 CPU를 가진 PC환경에서 GOP길이를 64정도로 길게 하여 Wyner-Ziv 부호화를 상대적으로 많이 수행하는 경우에 초당 700 ~ 2,300장을 부호화 할 수 있음을 확인했다. 제안방법은 현재 TDWZ 분산비디오부호화기에 적용하여 성능을 확인하였으나 LDPCA를 사용하는 픽셀영역 Wyner-Ziv(PDWZ, Pixel domain Wyner-Ziv) 분산비디오부호화 구조에도 쉽게 적용 가능할 것이다.

II. 기존 분산비디오부호화 방법

그림 1은 본 논문에서 사용한 분산비디오부호화기의 구조로써, 분산비디오부호화기의 현실적인 예인 변환영역 Wyner-Ziv(TDWZ, Transform domain Wyner-Ziv) 분산비디오부호화 방법을 나타낸다. TDWZ 분산비디오부호화기에서는 입력영상을 KEY 영상(K)과 Wyner-Ziv 영상(X)으로 분해한 후, KEY 영상은 H.264/AVC^[9]와 같은 기존의 부호화 방식으로 부호화 및 복호화 하고, Wyner-Ziv 영상은 변환(transform)/역변환(inverse transform), 양자화(quantization)/역양자화(inverse quantization), Slepian-Wolf 부/복호화, 양자화계수 복원(reconstruction) 등의 과정을 통해 부호화 및 복호화 한다^[3-8].

그림 1의 변환영역 Wyner-Ziv 영상의 부호화 과정은 다음과 같이 동작한다. 먼저 입력영상에 대해 소정의 변환과정("T"블록)을 수행하여 영상 신호의 에너지를 집중시킨다. 이를 위하여 본 논문에서는 정수 DCT 변환을 사용한다. 정수 DCT 변환계수는 변환 계수별로 2^T ,

$T \in \{1, 2, 3, \dots\}$ 레벨 양자화(“Q”블록)가 된다^[4-8]. Slepian-Wolf 부호화기는 양자화 된 정보를 비트플레인 단위로 분해한 후 각각의 비트플레인들을 비트율 조절이 가능한 LDPCA^[10, 11]를 사용하여 채널코드 부호화 한다. 그림 2는 정수 DCT와 Slepian-Wolf 부호화를 위해 순차적인 LDPCA 채널코드 부호화 방법을 사용하는 TDWZ 분산비디오부호화기의 부호화 방법을 상세하게 나타낸다. 이때 채널코드 부호화는 입력된 메시지가 출력 메시지에 포함되도록 하는 systematic 부호화 방법을 사용한다. 변환 및 양자화를 거친 정보를 비트플레인 단위로 구성하지 않고 블록마다 심벌 단위로 구성하는 것도 가능하지만 Wyner-Ziv 복호화 복잡도나 성능 측면에서 블록들의 심벌들을 여러 개의 비트플레인들로 구성하여 처리하는 것이 효과적이다^[13]. 이러한 부호화 동작은 기존의 예측부호화 방법과는 다르게 원본 영상 간에 독립적으로 수행된다는 점에서 기존의 인트라 부호화 방법과 유사하다. 그러나, 이러한 TDWZ 분산비디오부호화기는 그림 1의 TDWZ 분산비디오복호화기 구조에서 살펴볼 수 있듯이, 기 복원된 KEY 영상(\tilde{K})을 Wyner-Ziv 영상의 복원과정에서 이용함으로써 기존의 인트라 부호화 방법보다 더 우수한 복호화 성능을 보일 수 있다^[1-3].

Wyner-Ziv 영상(\tilde{X})의 복원을 위해서는 크게 두가지 정보를 이용하는데, 보조정보(Y)와 Slepian-Wolf 복원된 정보가 그것이다. 보조정보생성부(그림 1의 “Side informa-

tion generation” 블록)는 복원된 KEY 영상(\tilde{K})을 이용하여 원본 Wyner-Ziv 영상(X)과 유사하도록 보조정보(Y)를 만든다. 이때, 보조정보(Y)의 화질은 복호화 화질에 직접적인 영향을 미치므로, 기존의 프레임률 변환(frame rate conversion) 문제와 유사하게 KEY 영상 상호간 시간적 상관성을 이용하여 KEY 영상사이에 존재하는 영상에 대한 보조정보를 가능한 정확히 만들어 최종 복원된 영상의 화질을 향상시킨다^[14-16]. 하지만 복호화기에서 추가정보 없이 보조정보 화질을 향상시키는 데에는 한계가 있으므로 분산비디오복호화 과정에서는 보조정보(Y)와 더불어 Wyner-Ziv 부호화기에서 생성한 추가정보인 패리티정보를 전송 받아 복원

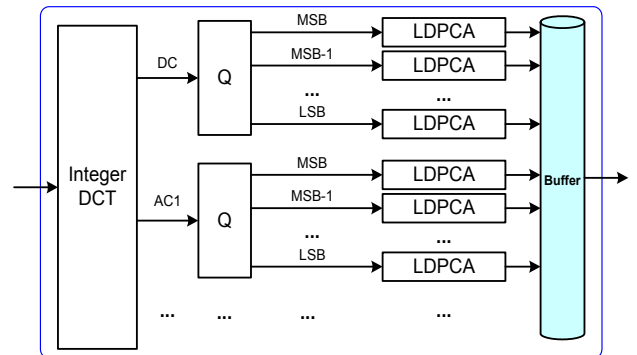


그림 2 . 기존의 순차적 LDPCA 채널코드 부호화 방법을 사용한 TDWZ 분산비디오부호화기
Fig. 2. Previous TDWZ distributed video encoder using sequential LDPCA encoding method

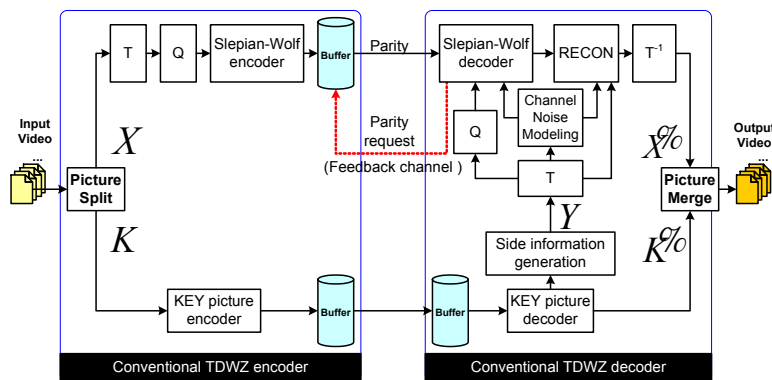


그림 1 . 변환영역 Wyner-Ziv 부호화기 구조
Fig. 1. Transform-domain Wyner-Ziv video codec structure

에 이용한다. 이를 위하여 Slepian-Wolf 복호화(“Slepian-Wolf decoder”블록) 및 양자화 계수의 복원(“RECON”블록) 과정을 수행한다^[3-8].

본 논문의 분산비디오부호화기는 원본 Wyner-Ziv 영상에 대한 패리티정보를 만들기 위하여 LDPCA 채널코드 부호화 방법 [10, 11]을 사용한다. LDPCA 채널코드는 블록 부호의 일종인 저밀도 패리티 체크(LDPC, Low density parity check) 부호를 비트율 제어가 가능하도록 만든 것이다. LDPCA 부호화를 비트플레인별로 순차적으로 수행하는 과정을 보다 상세히 기술하면 다음과 같다.

먼저, 영상이 N 개의 블록으로 나뉘고, 각 블록은 DCT와 양자화를 거쳐 전체 B 개의 비트플레인으로 구성되었다고 가정하자. 그러면 식 (1)과 같이 b 번째 비트플레인 메시지를 $\chi^{(b)}$ 로 표현할 수 있다.

$$\chi^{(b)} = \{x_1^{(b)}, x_2^{(b)}, \dots, x_i^{(b)}, \dots, x_N^{(b)}\}$$

where $x_i^{(b)} \in \{0,1\}$, $b \in \{1,2,\dots,B\}$, $i \in \{1,2,\dots,N\}$ (1)

여기서 $x_i^{(b)}$ 는 주어진 영상에 대한 b 번째 비트플레인의 i 번째 블록의 비트를 나타낸다. 이러한 b 번째 비트플레인 메시지 $\chi^{(b)}$ 에 식 (2)에 있는 패리티생성행렬 H 를 곱하면 식 (3)과 같이 b 번째 비트플레인에 대한 패리티 데이터 $p^{(b)} = \chi^{(b)}H$ 를 생성할 수 있다.

$$H = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1j} & \dots & h_{1M} \\ h_{21} & h_{22} & \dots & h_{2j} & \dots & h_{2M} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ h_{i1} & h_{i2} & \dots & h_{ij} & \dots & h_{iM} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ h_{M1} & h_{M2} & \dots & h_{Mj} & \dots & h_{MM} \end{bmatrix}$$

where $h_{ij} \in \{0,1\}$ (2)

여기서 M 은 생성한 패리티 비트플레인의 길이를 나타내며, 채널코드 코드율은 $r = N/(N+M)$ 으로 표현할 수 있다. 단, 본 논문에서는 채널코드 코드율로써 $r = 0.5$ ($N = M$)을 사용한다.

$$p^{(b)} = \chi^{(b)}H = \{p_1^{(b)}, p_2^{(b)}, \dots, p_j^{(b)}, \dots, p_M^{(b)}\}$$

where $p_j^{(b)} \in \{0,1\}$, $b \in \{1,2,\dots,B\}$, $j \in \{1,2,\dots,M\}$ (3)

따라서 식 (3)의 $p_j^{(b)}$ 는 b 번째 비트플레인의 j 번째 패리티비트를 나타내며, 식 (4)와 같이 입력 메시지들을 이에 대응하는 패리티 생성 행렬의 열벡터와 곱한 후 2진수로 만들기 위한 modulo 2의 연산을 수행하여 계산될 수 있다.

$$p_j^{(b)} = (x_1^{(b)}h_{1j} + x_2^{(b)}h_{2j} + \dots + x_i^{(b)}h_{ij} + \dots + x_N^{(b)}h_{Nj}) \text{mod}2$$
 (4)

즉, Slepian-Wolf 부호화는 양자화된 B 개의 메시지 비트플레인 $\chi^{(1)}, \dots, \chi^{(b)}, \dots, \chi^{(B)}$ 를 LDPCA 채널코드로 부호화하여 B 개의 패리티 비트플레인 $p^{(1)}, \dots, p^{(b)}, \dots, p^{(B)}$ 를 생성하는 과정이라 할 수 있다.

원본영상을 양자화한 후 전송해야 할 비트플레인의 수 (B)가 커질수록 LDPCA 채널코드 부호화해야 할 연산량도 증가하게 되는데, 이러한 B 값은 그림 3과 같이 부호화 시 사용하는 양자화 행렬에 의존하여 결정되며, 각각의 행렬에 대해 B 는 {10, 11, 17, 30, 36, 45, 50, 63}의 값을 갖는다. 이렇게 각 변환계수 위치 별로 채널코드 부호화할 비트플레인 수가 결정되는 이유는 그림 3의 2의 지수승 레벨 양자화 행렬들을 사용하여 영상의 모든 영역에 대해 동일하게 양자화를 수행하기 때문이다.

$$Q_1 = \begin{pmatrix} 16 & 8 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, Q_2 = \begin{pmatrix} 32 & 8 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, Q_3 = \begin{pmatrix} 32 & 8 & 4 & 0 \\ 8 & 4 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, Q_4 = \begin{pmatrix} 32 & 16 & 8 & 4 \\ 16 & 8 & 4 & 0 \\ 8 & 4 & 0 & 0 \\ 4 & 0 & 0 & 0 \end{pmatrix}$$

$$Q_5 = \begin{pmatrix} 32 & 16 & 8 & 4 \\ 16 & 8 & 4 & 4 \\ 8 & 4 & 4 & 0 \\ 4 & 4 & 0 & 0 \end{pmatrix}, Q_6 = \begin{pmatrix} 64 & 16 & 8 & 8 \\ 16 & 8 & 8 & 4 \\ 8 & 8 & 4 & 4 \\ 8 & 4 & 4 & 0 \end{pmatrix}, Q_7 = \begin{pmatrix} 64 & 32 & 16 & 8 \\ 32 & 16 & 8 & 4 \\ 16 & 8 & 4 & 4 \\ 8 & 4 & 4 & 0 \end{pmatrix}, Q_8 = \begin{pmatrix} 128 & 64 & 32 & 16 \\ 64 & 32 & 16 & 8 \\ 32 & 16 & 8 & 4 \\ 16 & 8 & 4 & 0 \end{pmatrix}$$

그림 3 . 양자화 행렬 [4-6]
Fig. 3. Quantization matrix [4-6]

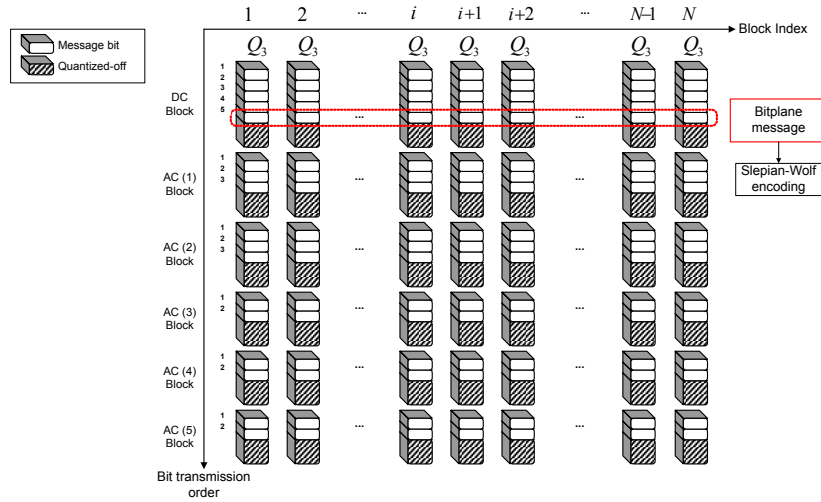


그림 4 . Wyner-Ziv 부호화 예
Fig. 4. Example of Wyner-Ziv encoding

그림 4는 양자화행렬 Q_3 을 사용하여 양자화 하는 Wyner-Ziv 부호화 방법을 나타낸다. 양자화 행렬 Q_3 을 사용하여 양자화 하는 경우 지그재그 스캔순서의 변환계수 위치별로 LDPCA 채널코드로 부호화해야 할 비트플레인들의 수는 {5, 3, 3, 2, 2, 2, 0, ..., 0}으로 결정된다. 이 경우 $B=17$ 로써, LDPCA 채널코드 부호화는 총 17회 수행되어야 한다. 마찬가지로 다른 양자화 행렬을 사용하면 LDPCA 채널코드 부호화에 소요되는 시간은 달라지며 B 값이 커질수록 많은 비트를 처리하여야 하여 LDPCA 채널코드 부호화에 소요되는 연산량도 비례적으로 증가한다. 이런 이유로 상

대적으로 고품질의 부호화를 수행하면 저화질로 부호화 하는 것에 비해 부호화 속도가 느려지는 문제가 발생한다. 따라서, 전송해야할 메시지가 많아지는 경우에도 분산비디오 부호화 방법을 사용하여 고속으로 부호화하기 위해서는 효과적인 채널코드 부호화 방법이 필요하다.

III. 병렬 LDPCA 채널코드 부호화를 이용한 TDWZ 부호화기 고속화

앞서 서술한 TDWZ 부호화기에서 LDPCA 채널코드 부

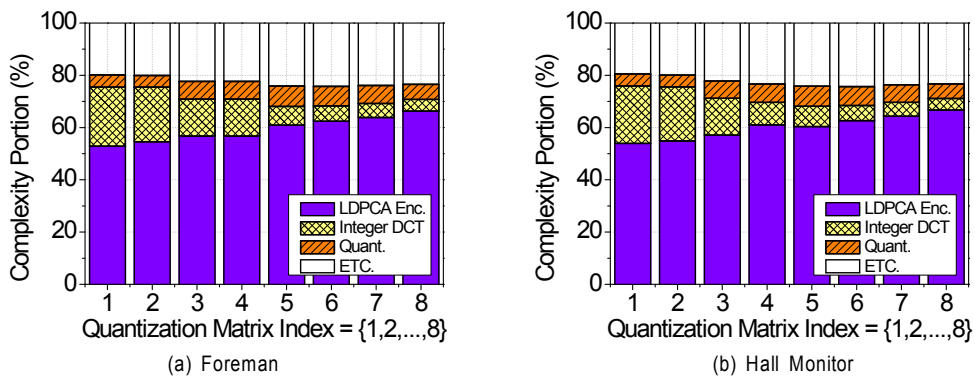


그림 5. 기존의 순차적 LDPCA 채널코드 부호화시 Wyner-Ziv 부호화기 요소별 계산시간 복잡도 비율
Fig. 5. Computing time Complexity of Wyner-Ziv encoding with sequential LDPCA method

호화에 소요되는 복잡도 증가 문제를 더욱 자세히 알아보기 위하여, 실제 분산비디오부호화 수행 시 LDPCA 채널코드 부호화가 차지하는 계산시간 비율을 측정하여 그림 5에 나타내었다. 이때 모든 수치는 100회 반복 실험하여 산술평균한 값을 사용하였다. 그림 5가 나타내는 바와 같이, 정수 DCT는 평균 11%, 양자화는 평균 6%, 기타 데이터 처리는 평균 22%의 부호화 시간을 차지하였으며, LDPCA 채널코드 부호화는 평균 60% 정도로써 Q_1 인 경우 54%, Q_8 인 경우 67%까지 증가하였다.

암달의 법칙(Amdahl's law)으로 알려진 시스템 성능 향상 원리에 의하면, 전체 시스템의 속도 향상을 위해서는 상대적으로 높은 시간 복잡도를 차지하는 요소를 먼저 고속화하는 것이 유리하다^[17]. 따라서 현재 TDWZ 부호화기를 고속화하기 위해서는 LDPCA 채널코드 부호화를 고속화하는 것이 중요하다.

한편, 그림 2에 표시한 LDPCA 채널코드 부호화 방법에서 관찰할 수 있는 특징은 LDPCA 채널코드 부호화할 데이터 상호간에 의존성이 존재하지 않는다는 점이다. 데이터 상호간에 의존성이 존재하지 않으면 일반적으로 단일 명령 복수 데이터(SIMD, single instruction multiple data) 처리 방식과 같은 병렬처리의 구현이 용이하다^[12]. 본 논문에서는 이러한 분산비디오부호화기의 특징에 주목하여 그림 6과 같은 병렬 LDPCA 채널코드 부호화 방법을 제안한다. 제안 방법은 다음과 같이 2가지 단계를 거쳐서 수행된다.

- (단계 1) 양자화된 비트플레인들을 하나의 메시지로 묶음 (그림 6의 “Pack Data” 블록)
- (단계 2) 묶어진 비트플레인 메시지들을 병렬로 LDPCA 부호화 수행 (“Parallel LDPCA” 블록)

단계 1은 양자화된 비트플레인 메시지들을 하나로 묶는 것으로써, 그 세부과정은 다음과 같다. 먼저 B 개의 채널코드 부호화해야 할 비트플레인들이 존재한다고 가정하자. 1부터 B 번째까지의 비트플레인들을 하나의 메시지 집합으로 구성하는 것이 그림 6의 “Pack Data”블록의 역할이다. 식 (5)는 본 논문에서 사용한 메시지 집합을 구성하는 방법으로써, 1부터 B 번째까지의 비트플레인들의 비트들을 2의

승수를 바꿔가면서 곱하여 X_i 라고 하는 새로운 메시지를 구성한다.

$$X = \{X_1, \dots, X_i, \dots, X_N\},$$

$$X_i = x_i^{(1)}2^{B-1} + x_i^{(2)}2^{B-2} + \dots + x_i^{(b)}2^{B-b} + \dots + x_i^{(B)} \quad (5)$$

where $x_i^{(b)} \in \{0,1\}$, $b \in \{1,2,\dots,B\}$

여기서 $x_i^{(b)}$ 는 b 번째 비트플레인의 i 번째 블록의 비트 값이다. 주목할 점은 X_i 는 i 번째 블록에 포함된 1부터 B 번째까지의 모든 비트플레인들의 정보를 포함하고 있다는 점이다. 그러나 이렇게 메시지를 구성함에 있어서 반드시 식 (5)와 같은 순서로 2의 승수를 곱할 필요는 없다.

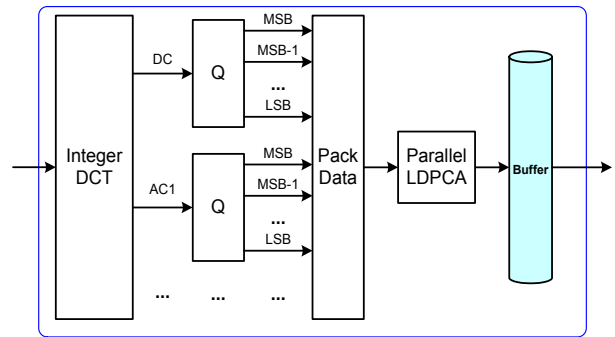


그림 6. 제안하는 병렬 LDPCA 채널코드 부호화 방법을 사용한 Wyner-Ziv 부호화기
 Fig. 6. Proposed Wyner-Ziv video encoder using parallel LDPCA encoding method

예를 들어 식 (5)에서는 최상위 비트(Most significant bit)인 $x_i^{(1)}$ 에 2^{B-1} 를 곱했으나, 만약 다른 비트들끼리 서로 겹치지 않음으로 인해서 전체 비트플레인들의 정보가 유지된다면 어떠한 조합이든 가능하다. 왜냐하면 이렇게 구성된 병렬처리용 메시지는 한 번의 연산으로 여러 개의 데이터가 동시에 연산됨을 가정하므로 배열하는 순서 자체는 중요하지 않기 때문이다. 이렇게 구성된 메시지는 그림 6의 “Parallel LDPCA”블록을 통해 병렬처리 채널코드 부호화된다. 이는 단일 명령 복수 데이터(SIMD, single instruction multiple data) 처리 방법으로 알려진 병렬화 방법^[12]으로 분류할 수 있으며 부호화해야 할 비트플레인 메시지들 간에

표 1. 기존의 순차적 LDPCA 채널코드 부호화와 제안하는 병렬 LDPCA 채널코드 부호화 코드의 예
 Table 1. LDPCA encoding by previous sequential LDPCA and the proposed parallel LDPCA method

(a) 기존의 순차적 LDPCA 부호화 코드	(b) 제안하는 병렬 LDPCA 부호화 코드
<pre> void encode_serial_ldpca(int32_t *bin_src, int32_t *bin_accumsdr) { int32_t k, t; for(k = 0; k < m; k++) bin_accumsdr[k] = 0; // Source * H' for(k = 0; k < n; k++) for(t = jc[k]; t < jc[k+1]; t++) bin_accumsdr[ir[t]] += bin_src[k]; // Accumulation for(k = 1; k < m; k++) bin_accumsdr[k] += bin_accumsdr[k-1]; // Modulo 2 for(k = 0; k < m; k++) bin_accumsdr[k] = bin_accumsdr[k] % 2; } </pre>	<pre> void encode_parallel_ldpca(uint64_t *packed_src, uint64_t *packed_accumsdr) { int32_t k, t; for(k = 0; k < m; k++) packed_accumsdr[k] = 0; // Source * H' for(k = 0; k < n; k++) for(t = jc[k]; t < jc[k+1]; t++) { packed_accumsdr[ir[t]] = packed_accumsdr[ir[t]] ^ packed_src[k]; } // Accumulation for(k = 1; k < m; k++) packed_accumsdr[k] ^= packed_accumsdr[k-1]; } </pre>

상호 의존성이 존재하지 않기 때문에 가능한 것이다.

표 1은 기존의 순차적 LDPCA 채널코드 부호화와 제안하는 병렬 LDPCA 채널코드 부호화를 C언어로 표현한 것이다. 순차적 LDPCA 채널코드 부호화를 수행하는 전체 소스 코드는 [11]에서 이용할 수 있다. 표 1 (a)에서 32비트(혹은 8비트) 정수형 변수의 배열 bin_src를 통해 입력된 양자화된 비트플레인 메시지는 채널코드 부호화 원리에 의해 LDPCA 채널코드 부호화되어 32비트 정수형 변수의 배열 bin_accumsdr을 통해 출력된다. 이때 각각의 메시지 비트들은 각각의 32비트 정수형 변수에 적재되고 연산이 수행된다. 반면 표 1 (b)에서는 64비트 정수형 변수의 배열 packed_src에 여러 개의 비트플레인 메시지들이 묶여져서 입력되고, 동일 크기의 변수인 packed_accumsdr에 패리티 비트플레인들이 묶여져서 출력된다. 그림 3의 양자화 행렬을 사용하는 경우 B의 최대값은 63이므로 64비트 정수형 변수는 그림 3에 나타난 모든 양자화 행렬에 대응할 수 있다. 표 1 (b)는 표 1 (a)와 다르게 덧셈을 사용하지 않고 비트별 XOR(Bitwise exclusive OR) 연산을 사용하고 있는데, 이는 덧셈 연산으로 인해 자리올림(Carry)이 발생하지 않으면서도 이진수의 덧셈을 수행할 수 있기 때문이다. 그리고

비트별 XOR 연산을 사용함으로써 각각의 비트들은 이진수 형태를 계속 유지하므로 modulo 2 연산을 수행할 필요가 없는 장점이 있다.

IV. 실험결과

제안방법의 성능확인을 위해 모든 실험은 Intel® Pentium® E5200@2.50GHz CPU, 2 Gbytes RAM, Microsoft Windows 7 운영체제 상에서 수행하였다. 실험영상으로는 QCIF@15Hz 해상도의 Foreman과 Hall monitor 영상 각각에 대해 총 149장씩 사용하였다. 소요 시간 및 복잡도의 측정을 위해 100회 반복 부호화 실험 후 소요 시간 및 부호화 성능의 산술평균을 산출하였다. 제안하는 고속 분산비디오 부호화기에서 KEY 영상은 H.264/AVC를 고속으로 구현하는 공개소스 프로젝트인 x264^[18]를 H.264/AVC baseline profile로 설정 한 후 모두 인트라 부호화하였다. 비트율 제어 위해 Wyner-Ziv 부호화는 그림 3의 8가지 양자화 행렬을 사용하였고 이에 대응하는 KEY영상의 양자화 파라미터는 Foreman영상에 대해서는 QP={40, 39, 38, 34, 34, 32,

29, 25}, Hall monitor에 대해서는 $QP=\{37, 36, 36, 33, 33, 31, 29, 24\}$ 를 사용하였다⁶⁾.

그림 7은 기존의 순차적 LDPCA 채널코드 부호화와 제안하는 병렬 LDPCA 채널코드 부호화를 각각 수행하여 영상 1장의 부호화에 소요되는 시간을 비교한 것이다. 기존방법으로는 Q_1 인 경우 218 us, Q_8 에서는 1,371 us까지 소요되었으나, 제안방법에서는 Q_1 에서 Q_8 까지 부호화 시간이 25 us로 거의 일정하게 유지되는 결과를 보였다. 이는 그림 2와 같은 기존의 방법에서는 고비트율로 갈수록 처리해야할 데이터가 증가하고 이에 비례하여 LDPCA 채널코드 부호화를 위한 연산량 증가가 발생했으나, 그림 6과 같은 제안방법은 처리해야할 데이터들을 하나의 메시지묶음(Packed message)으로 처리함으로써 LDPCA 채널코드 부

호화에 소요되는 연산량이 모든 비트율에서 거의 동일하게 유지되었기 때문이다. 기존 방법에 비해 제안하는 병렬 LDPCA 부호화 방법은 부호화 시간이 1/8 ~ 1/55까지 절감됨을 확인할 수 있다. 이런 이유로 인해 그림 8과 같이 TDWZ 부호화기에서 LDPCA 채널코드 부호화를 위해 소요되는 복잡도의 비율은 평균 9% 정도로 낮아지게 되었다. 그러나 이러한 속도 향상에도 불구하고 GOP=2에서 영상 전체를 부호화 할 때의 부호화 속도는 GOP=1(모든 영상을 KEY영상 부호화)에 비해서 2배 이상 빠를 수 없다. 이는 식 (6)과 같이 압달의 공식으로 계산할 수 있는데¹⁷⁾, Wyner-Ziv 부호화 부분이 아무리 빨라지더라도 기존의 KEY 영상의 부호화 속도가 더 이상 개선되지 않는다면 더 이상의 속도향상은 발생할 수 없기 때문이다.

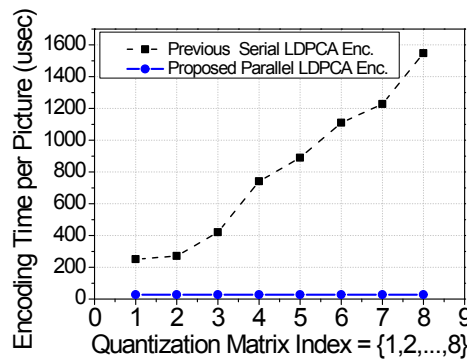
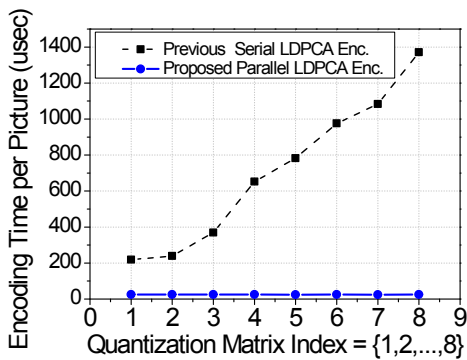


그림 7. 순차적 LDPCA 채널코드 부호화와 제안하는 병렬 LDPCA 채널코드 부호화에 걸리는 시간비교
 Fig. 7. Encoding time comparison of the previous sequential and the proposed parallel LDPCA encoding methods

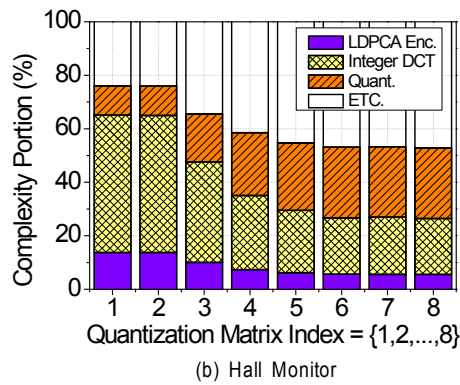
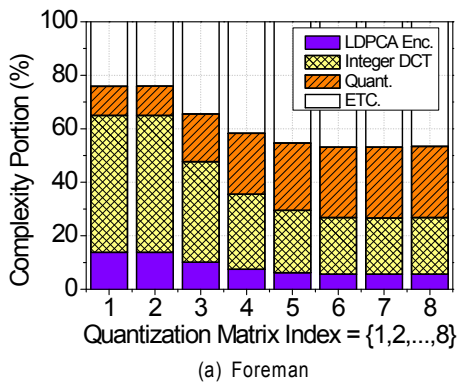


그림 8. 제안하는 병렬 LDPCA 채널코드 부호화시 Wyner-Ziv 부호화기 요소별 복잡도 비율
 Fig. 8. Computation complexity of Wyner-Ziv encoder with proposed parallel LDPCA encoding method

$$\begin{aligned}
 P_{KEY} &= 0.5, P_{WZ} = 0.5, \\
 speedup &= \lim_{S_{WZ} \rightarrow \infty} \left(\frac{1}{(1 - P_{WZ}) + \frac{P_{WZ}}{S_{WZ}}} \right) \\
 &= \lim_{S_{WZ} \rightarrow \infty} \left(\frac{1}{(1 - 0.5) + \frac{0.5}{S_{WZ}}} \right) = 2
 \end{aligned} \tag{6}$$

여기서 P_{KEY} 와 P_{WZ} 는 각각 GOP내에서 부호화해야 할 KEY 영상과 Wyner-Ziv 영상의 비율, S_{WZ} 는 Wyner-Ziv 영상의 부호화의 고속화 정도, $speedup$ 은 TDWZ 분산비디오부호화기 전체의 속도 향상을 나타낸다. 이런 문

제는 GOP의 수를 증가시킴으로써 해결할 수 있는데, 본 논문에서는 제안 방법의 속도 향상 정도에 대한 실험을 위해 GOP를 {2, 4, 8, 64}로 변화시키면서 기존 방법과 제안방법에 따른 부호화 속도를 비교하였다.

그림 9는 이에 대한 결과로써, 제안하는 병렬 LDPCA 채널코드 부호화 방법은 기존의 순차적으로 LDPCA 채널코드 부호화를 수행하는 방법에 비해 전반적으로 속도가 개선되고 있으며, GOP길이가 64정도로 길어지는 경우와 같이 KEY영상보다는 Wyner-Ziv 영상으로 부호화 되는 영상의 수가 증가하게 되면 초당 700 ~ 2,300 장까지 영상의 부호화가 가능하게 됨을 확인할 수 있다. 단, GOP 길이는

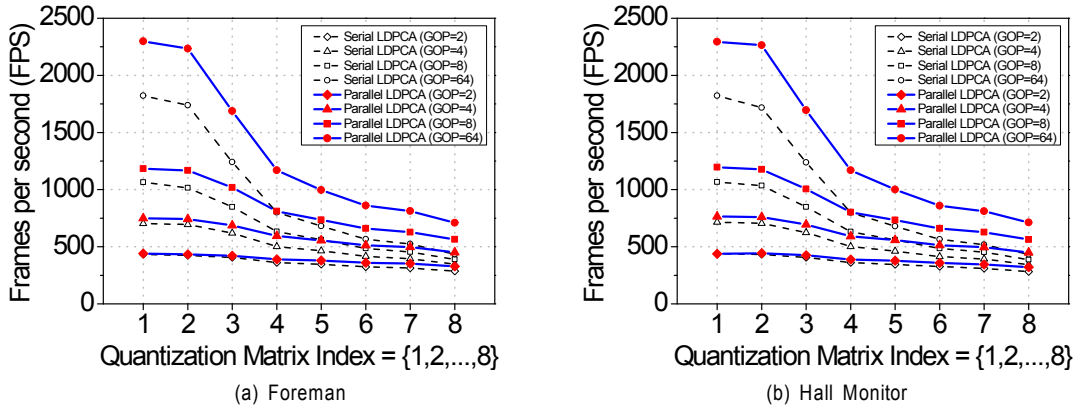


그림 9. GOP 크기에 따른 부호화 속도 비교, GOP={2, 4, 8, 64}
 Fig. 9. Encoding speed comparison according to the GOP size, GOP={2, 4, 8, 64}

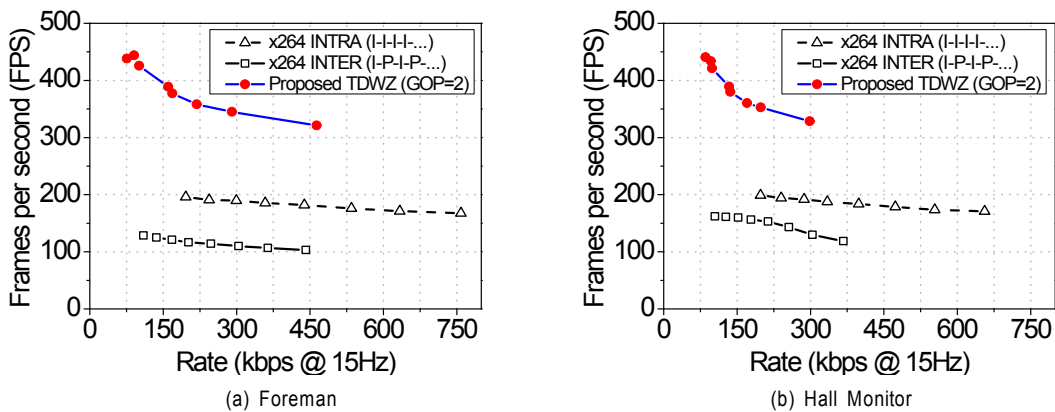


그림 10 . GOP=2 일 때, x264 부호화기와 제안방법 사이의 부호화 속도 비교
 Fig. 10. Encoding speed comparison between proposed fast Wyner-Ziv encoder and x264 encoder in GOP=2

영상의 임의 접근(Random access)능력과 입력 영상의 특성에 맞춰 결정되어야 한다. 왜냐하면 빠른 움직임 특성을 갖는 영상의 경우 GOP 길이가 길어지게 되면 비트율-왜곡 성능이 크게 감소할 수 있기 때문이다. 그림 9에서 GOP=2인 경우는 Wyner-Ziv부호화기의 속도가 상대적으로 가장 느리지만, 기존의 비디오 부호화 기술에 비하면 매우 빠른 속도임을 그림 10을 통해 확인할 수 있다.

그림 10의 실험결과는 H.264/AVC의 고속구현 코덱인 x264 코덱^[18]으로 인트라(I-I-I-I...) 및 인터 부호화(I-P-I-P...)한 결과와 GOP=2를 갖는 제안방법 사이의 성능을 비교한 것이다. 제안방법은 x264의 인트라 부호화에 비해서는 약 2배, 인터 부호화에 비해서는 약 3 배 정도 부호화 속도가 빠른 것으로 나타났다.

V. 결 론

본 논문에서는 변환영역 Wyner-Ziv 분산비디오부호화기에서 가장 많은 연산 복잡도를 보이는 LDPCA 부호화기를 일종의 단일 명령 복수 데이터 처리 방식으로 병렬화 함으로써 LDPCA 부호화 속도를 8 ~ 55배까지 향상 시켰으며 이를 통해 Wyner-Ziv 영상의 부호화 속도도 크게 향상시킬 수 있었다. 제안 방법은 다양한 하드웨어 아키텍처에서 이용 가능하고, 변환영역이 아니더라도 픽셀영역에서 LDPCA 채널코드 부호화를 사용하는 Wyner-Ziv 분산비디오부호화 환경에도 쉽게 적용 가능할 것이다.

참 고 문 헌

[1] J. Slepian and J. Wolf, "Noiseless coding of correlated information sources," IEEE Trans. Inf. Theory, pp.471-480, 1973.
 [2] A. Wyner and J. Ziv, "The rate-distortion function for source coding

with side information at the decoder," IEEE Trans. Inf. Theory, pp.1-10, 1976.
 [3] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed Video Coding," Proceedings of the IEEE, pp. 71-83, 2005.
 [4] A. Aaron, S. Rane, E. Setton, and B. Girod, "Transform domain Wyner-Ziv codec for video," Proc. Visual Communications and Image Processing, VCIP2004, Jan. 2004.
 [5] C. Brites, J. Ascenso, F. Pereira, "Improving transform domain Wyner-Ziv video coding performance," IEEE International Conference on Acoustics, pp. 525-528, May 2006.
 [6] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, M. Ouaret, "The discover codec: architecture, techniques and evaluation," Picture Coding Symposium, 2007.
 [7] J. Park, B. Jeon, D. Wang, and A. Vincent, "Wyner-Ziv Video Coding with Region Adaptive Quantization and Progressive Channel Noise Modeling", IEEE Broadband Multimedia Systems and Broadcasting 2009, pp. 1-6, May 2009.
 [8] 박종빈, 심혁재, 전병우, "블록 적응 양자화를 이용한 Wyner-Ziv 비디오 코딩", 2008 신호처리 합동학술대회, 제21권 제 1호, pp.37, 2008.
 [9] I.E.G. Richardson, H.264 and MPEG-4 Video Compression: Video Coding for Next-generation, Wiley 2003.
 [10] D. Varodayan, A. Aaron, and B. Girod, "Rate-Adaptive Codes for Distributed Source Coding," EURASIP Signal Processing Journal Special Issue on DSC 86(11), pp.3123-3130, 2006.
 [11] LDPCA C source (<http://www.stanford.edu/~divad/software/ldpca.zip>).
 [12] M. Flynn, "Some Computer Organizations and Their Effectiveness," IEEE Transaction on Computers, Vol. 21, No. C., pp. 948-960, 1972.
 [13] R.P. Westerlaken, S. Borchert, R.K. Gunnewiek, R.L. Lagendijk, "Analyzing Symbol and Bit Plane-Based LDPC in Distributed Video Coding," ICIP 2007, pp. II-17 - II-20, 2007.
 [14] 최영태, 양시영, 정제창, "MPEG 비디오 트랜스코딩에서 프레임율 변환을 위한 효율적인 움직임 벡터 재추정 기법," 한국방송공학회, 2004년도 한국방송공학회 정기총회 및 학술대회, pp. 233-236, 2004년 11월.
 [15] 민경연, 박시내, 심동규, "계층적 적응적 탐색과 양방향 움직임 예측을 이용한 프레임율 증가 방법," 대한전자공학회, 전자공학회논문지-SP, pp. 28-36, 2009년 5월.
 [16] P. Blanchfield, D. Wang, A. Vincent, F. Speranza, R. Renaud, "Advanced Frame Rate Conversion and Performance Evaluation," SMPTE Motion Imaging Journal, pp. 153-159, April 2006.
 [17] D. A. Patterson and J. L. Hennessy, Computer Organization and Design, Third Edition, Morgan Kaufmann Publishers, pp. 40-42, 1998.
 [18] x264 open source project (<http://www.videolan.org/developers/x264.html>).

저 자 소 개



박 종 빈

- 2004년 : 성균관대학교 정보통신공학부 졸업(학사)
- 2006년 : 성균관대학교 전자전기공학과 졸업(석사)
- 2006년 ~ 현재 : 성균관대학교 전자전기컴퓨터공학과 박사과정
- 주관심분야 : 분산비디오부호화, 다시점비디오부호화, 영상인식, 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 선임/수석연구원
- 1997년 ~ 현재 : 성균관대학교 정보통신공학부 교수
- 주관심분야 : 멀티미디어 영상압축, 영상인식, 신호처리