논문 14-39A-11-01 The Journal of Korean Institute of Communications and Information Sciences '14-11 Vol.39A No.11
http://dx.doi.org/10.7840/kics.2014.39A.11.625

웨이브렛 계수의 비트율-왜곡 최적화 기반 블록 부호화를 이용하는 임베디드 영상 압축 방법

양 창 모◆, 정 광 수°

Embedded Image Compression Scheme Using Rate-Distortion Optimized Block Coding of Wavelet Coefficients

Chang Mo Yang◆, Kwangsue Chung°

요 약

본 논문에서는 웨이브렛 계수의 비트율-왜곡 최적화 기반 블록 부호화를 사용하는 새로운 임베디드 영상 압축 방법을 제안한다. 웨이브렛 계수의 크기에 따라 셋분할 부호화나 블록분할 부호화를 수행하는 기존의 임베디드 부호화 방법들과는 달리, 제안한 방법에서는 웨이브렛 계수나 블록들을 비트율-왜곡비 기댓값에 따라 정렬함으로서 비트율-왜곡 최적화를 수행하는 동시에 비트율-왜곡비 기댓값을 이용하여 최적화된 블록분할 부호화를 수행한다. 또한 제안한 방법에서는 엔트로피 부호화를 위해 웨이브렛 계수들간에 존재하는 다양한 상관관계를 이용한다. 실험 결과는 제안한 영상 압축 방법이 기존의 임베디드 부호화 방법인 SPIHT 및 EBCOT와 비교하여 각각 0.11~1.16dB 및 -0.18~0.52dB의 PSNR 성능향상을 제공함으로써 평균적으로 우수한 성능을 제공함을 보여준다.

Key Words : Wavelet, Embedded, Image Compression, Block Coding, RDBC

ABSTRACT

In this paper, we propose a new embedded image compression scheme which uses rate-distortion optimized block coding of wavelet coefficients. Unlike to previous works in which set-partition or block-partition is performed according to the magnitude of wavelet coefficients, the proposed scheme achieves rate-distortion optimization by sorting wavelet coefficients or blocks according to their expected rate-distortion slope. At the same time, it performs the optimized block-partition coding using the expected rate-distortion slope of blocks. The proposed scheme also uses various relationship of wavelet coefficients for the entropy coding. Experimental results demonstrate that the proposed image compression scheme provides better overall performance than the existing embedded coding schemes such as SPIHT and EBCOT, in which the PSNR gains of the proposed scheme are about 0.11~1.16dB and -0.18~0.52dB, respectively.

Ⅰ. 서 론

최근, 멀티미디어 서비스에 대한 관심이 높아지면서 영상을 보다 효율적으로 압축하거나 전송하는 방법들이 활발히 연구되고 있다[1-3]. 특히, 최신의 정지영상 압축 표준인 JPEG2000에서 기존의 H.26x 권고

※ 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행되었습니다.[14-824-09-002, (2세부) 대규모 실시간 비디오 분석에 의한 전역적 다중 관심객체 추적 및 상황 예측 기술 개발]

◆ First Author : Smart Media Research Center, Korea Electronics Technology Institute, cmyang@keti.re.kr, 정회원
° Corresponding Author : Department of Communications Engineering, Kwangwoon University, kchung@kw.ac.kr, 종신회원
논문번호 : KICS2014-08-313, Received August 19, 2014; Revised November 11, 2014; Accepted November 11, 2014

안, MPEG 및 JPEG 표준에서 사용하던 이산 여현 변환(Discrete Cosine Transform, DCT)을 대신하여 이산 웨이브렛 변환(Discrete Wavelet Transform, DWT)을 채택한 후로 DWT를 이용한 영상 압축 기술이 주목받고 있다.

기존의 DCT 기반 영상 압축 방법에서는 영상을 일정한 크기의 블록으로 분할하여 주파수 변환과 양자화를 수행하기 때문에, 높은 비트율에서는 우수한 성능을 나타내는 반면 낮은 비트율에서는 블록화 현상(Blocking artifacts)이 발생한다. 반면에 DWT 기반 영상 압축 방법은 입력되는 영상 신호를 부대역으로 분해한 후 웨이브렛 계수들을 비트평면 단위로 부호화하기 때문에, 다중 해상도 표현과 계층적 양자화가 가능할 뿐만 아니라 임베디드 비트스트림을 만들기에도 적합하다. 또한 기존의 DCT 기반 영상 압축 방법들과 비교하여 낮은 비트율에서도 블록화 현상이 발생하지 않는다.

일반적으로 DWT를 이용하는 임베디드 영상 압축 방법[4-11]에서는 웨이브렛 계수들을 셋(Set)이나 블록(Block)으로 구성한 후, 지수적으로 감소하는 크기 임계값을 이용하여 셋이나 블록의 중요도를 판별하고 판별된 중요도에 따라 셋이나 블록을 연속적으로 분할하는 셋분할 부호화나 블록분할 부호화가 사용된다. 이때 크기 임계값을 2의 승수의 값으로 선택함으로서 비트평면 단위의 부호화가 수행되어 부호화나 복호화 과정이 어떤 비트율에서 중단되더라도 전체 영상을 복원할 수 있는 임베디드 특징을 가지게 된다. 그러나 이러한 영상 압축 방법들은 웨이브렛 계수들을 크기에 따라 정렬하기 때문에 임베디드 비트스트림을 만들기에는 적합한 반면, 비트율-왜곡의 관점에서 최적화되어 있지 않은 단점이 있다. 이러한 단점을 개선하고자 일반적인 임베디드 부호화 방법을 변형하여 비트율-왜곡 최적화를 수행하는 연구들[12-14]이 있었으나, 기존 연구들에서는 비트율-왜곡 최적화를 위해 압축 비트스트림의 임베디드 특성을 희생시키거나[12] 웨이브렛 계수들간에 존재하는 상관관계를 충분히 활용하지 못하는 단점이 있다[13,14].

본 논문에서는 기존의 연구들이 가지는 단점들을 개선하기 위해 "비트율-왜곡 최적화 기반 블록 부호화(Rate-Distortion optimized Block Coding, RDBC)"를 이용하는 새로운 임베디드 영상 압축 방법을 제안한다. 압축 비트스트림의 임베디드 특징을 유지하면서도 비트율-왜곡 최적화를 수행하기 위해서, RDBC에서는 웨이브렛 계수나 블록들을 비트평면 단위로 부호화 했을 때 얻을 수 있는 비트율-왜곡비 기댓값(Expected rate-distortion slope)을 계산한 후, 계산된 비트율-왜곡비 기댓값을 연속적으로 감소하는 비트율-왜곡 임계값과 비교함으로서 웨이브렛 계수나 블록들을 정렬하는 한편 비트율-왜곡비 기댓값에 최적화된 블록분할 부호화를 수행한다. 또한, 웨이브렛 계수들간에 존재하는 다양한 상관관계를 엔트로피 부호화(Entropy coding)에 적용한다. 이러한 과정으로 생성되는 비트스트림은 일반적인 임베디드 부호화 방법들과 마찬가지로 비트평면 단위의 부호화 과정을 가짐으로, 부호화나 복호화 과정이 어떤 비트율에서 중단되더라도 전체 영상을 복원할 수 있는 임베디드 특징을 유지한다.

본 논문의 구성은 다음과 같다. 먼저 Ⅱ장에서는 웨이브렛 계수의 특징과 기존의 임베디드 압축 방법에 대해 살펴본다. Ⅲ장에서는 본 논문에서 제안된 RDBC에 대해 기술한다. RDBC는 6가지 단계의 기능 블록을 이용하여 영상을 부호화한다. 마지막으로 Ⅳ장 및 Ⅴ장에서는 기존의 임베디드 영상 압축 방법과 RDBC 기반 영상 압축 방법의 성능을 평가하고 결론을 맺는다.

Ⅱ. DWT 및 임베디드 영상 압축

2.1 DWT 및 웨이브렛 계수의 특징

DWT는 영상 신호를 부대역으로 분해하는 변환방식으로서 QMF(Quadrature Mirror Filter)[15]를 이용하여 구현된다. 그림 1에서 보듯이 입력되는 영상에 DWT를 수행하면 수직과 수평 방향으로 저대역 필터 L과 고대역 필터 H를 각각 한번씩 적용하고 2:1의 부표본화(Subsampling)를 수행함으로서, 세가지의 방향 선택적인 고주파 대역 LH, HL, HH와 저주파 대역 LL을 얻는다. 이러한 과정을 통해 얻어지는 가장 낮은 주파수 대역인 LL에는 대부분의 영상 에너지가 몰려 있으므로, 앞서 설명한 부대역 분해과정을 LL에

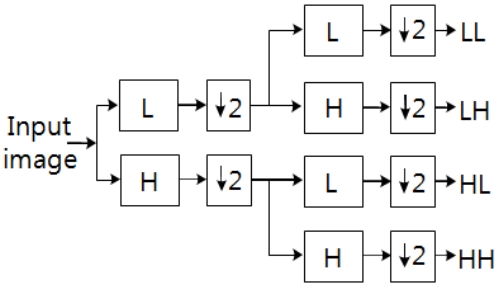


그림 1. DWT를 이용한 영상의 부대역 분해
Fig. 1. Subband decomposition of image using DWT

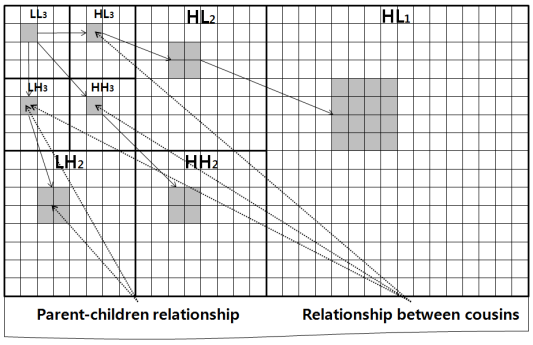


그림 2. 웨이브렛 계수들의 대역간 상관관계
Fig. 2. Inter-band correlation of wavelet coefficients

반복적으로 적용하여 다음 계층의 대역을 얻는다.

일반적으로 영상에 DWT를 수행하면 인접한 거리에 있는 계수들간에 상관관계를 갖는다. 한편, 대부분의 웨이브렛 계수들은 아주 작은 값을 가지게 되는 반면 일부 웨이브렛 계수들만이 입력 영상의 에지영역에 분포하며 큰 값을 가지는 통계적 특징인 지역화 특성(Localization property)[15]이 나타난다. 또한 웨이브렛 계수들은 그림 2와 같은 대역간 상관관계를 가진다. 이러한 상관관계는 나무구조(Tree structure)를 이용해서 나타내는데, 나무구조의 상위 계층의 노드를 부모(Parent) 노드라고 정의하고, 부모 노드의 바로 아래 계층에 존재하는 계수들을 자식(Children) 노드라고 정의하며, 이러한 대역간 웨이브렛 계수들의 상관관계를 부모-자식간 상관관계(Parent-children relationship)[5,6]로 정의한다. 이와 유사하게, 웨이브렛 계수들은 같은 크기의 다른 방향에 존재하는 대역간에도 상관관계를 가진다. 나무구조를 기준으로 이러한 계수들을 사촌(Cousin) 노드라고 정의하며, 이러한 대역간 상관관계를 사촌간 상관관계(Relationship between cousins)[16,17]로 정의한다.

2.2 임베디드 영상 압축 방법

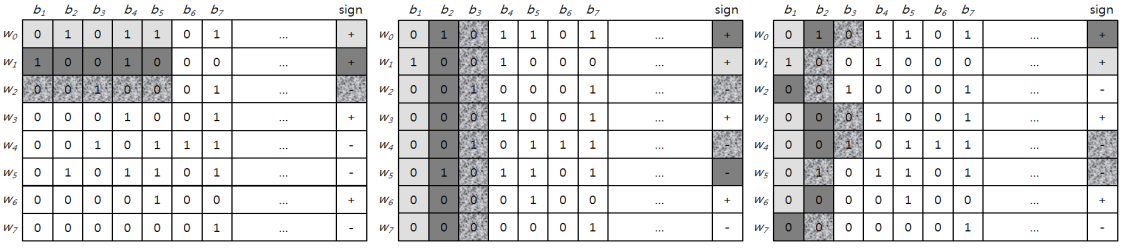
JPEG이나 MPEG가 같은 기존의 전통적인 영상 압축 방법에서는 입력되는 영상에 주파수 변환을 수행한 후, 변환 계수들을 심벌(Symbol) 단위로 부호화한다. 반면에 임베디드 영상 압축 방법에서는 변환 계수들을 2의 승수로 표현되는 크기 임계값과 비교하여 중요성 여부를 0 또는 1의 이진비트로 부호화하는 이진 비트평면 부호화 방법을 사용한다. 그림 3(a) 및 3(b)는 전통적인 영상 압축 방법과 임베디드 부호화 방법의 차이점을 보여준다. 그림 3에서 w_i는 변환계수들을 나타내며, b_i는 각 변환계수의 이진 비트를 0 또는 1의 값으로 표현한 비트를 나타낸다. 그러나 단순히 변환계수의 비트평면을 부호화하는 방법에서는 엔트로피 부호화(Entropy coding)를 제외하면 압축효과가 발생하지 않는다. DWT 기반의 임베디드 부호화 방법에서는 이러한 문제점을 웨이브렛 계수의 특징을 이용하여 해결하는데, 일반적으로 웨이브렛 계수를 셋이나 블록들의 집합으로 구성한 후, 셋이나 블록들을 지수적으로 감소하는 크기 임계값과 비교하여 이진 비트 혹은 적은 개수의 심벌로 부호화하는 방식을 사용함으로서 압축 효과를 발생시킨다.

DWT를 이용하는 대표적인 임베디드 부호화 방법으로는 셋분할 부호화 방법[5-8], 블록분할 부호화 방법[9-11], RDE(Rate-Distortion optimized Embedding)[13], EBCOT(Embedded Block Coding with Optimized Truncation)[14] 등이 있다. 셋분할 부호화 방법에서는 지수적으로 감소하는 크기 임계값과의 비교를 통해 웨이브렛 계수의 중요성 지도(Significance map)를 구성한 후, 부모-자식간 상관관계를 이용하여 제로트리(Zerotree)를 형성하고 부호화하는 방법을 사용한다.

블록분할 부호화 방법은 셋분할 부호화 방법이 대역간 상관관계를 이용하기 때문에 영상 부호화에 많은 메모리가 사용된다는 점을 개선한 방법으로서, 웨이브렛 계수의 지역화 특성을 이용한다. 블록분한 부호화 방법에서는 웨이브렛 계수들을 일정한 크기의 블록으로 구성한 후, 지수적으로 감소하는 크기 임계값에 따라 블록의 중요도를 이진 비트로 부호화하며, 해당 블록이 중요하다고 판단되는 경우 해당 블록을 4개의 부블럭(Sub-block)으로 분할하는 과정을 반복함으로서 영상을 부호화한다.

RDE는 기존의 임베디드 영상 압축 방법과 차별되는 웨이브렛 계수의 새로운 정렬방식을 사용한다. 그림 3(c)에서 보듯이, 기존의 방법에서 웨이브렛 계수들을 크기에 따라 정렬하면서 셋이나 블록분할을 통해 영상을 압축했던 것과는 달리 RDE에서는 셋이나 블록분할의 방법을 사용하지 않고 단순히 웨이브렛 계수들을 비트율-왜곡비 기댓값에 따라 정렬하고 엔트로피 부호화하는 방법을 사용한다.

JPEG2000 표준으로 채택된 EBCOT는 블록분할 부호화 방법과 RDE의 비트율-왜곡비 정렬 방법의 개념에서 착안한 영상 부호화 방법으로서, 기존의 임베디드 영상 부호화 방법들과 비교하여 매우 우수한 압축 성능을 제공한다. EBCOT는 두가지 단계로 구성되는데, 첫 번째 단계에서는 웨이브렛 계수들을 일정한 크기의 코드블록(Code-block)으로 분할한 후, 각 코드블록을 독립적으로 블록분할 부호화함으로서 다수개의 임베디드 비트스트림을 생성한다. 두 번째 단계에



(a) 전통적 부호화 방식
(a) Conventional coding

(b) 임베디드 부호화 방식
(b) Embedded coding

(c) RDE 부호화 방식
(c) RDE coding

그림 3. 여러 가지 부호화 방법 비교
Fig. 3. Comparisons of various coding methods

서는 임의의 지점에서 잘림이 발생해도 복호화가 가능하다는 임베디드 비트스트림의 특징과 비트율-왜곡 최적화 이론을 결합한 PCRD(Post Compression Rate-Distortion) 최적화를 수행함으로서, 생성된 다수개의 임베디드 비트스트림을 재조합하여 하나의 비트스트림을 생성한다.

Ⅲ. 비트율-왜곡 최적화 기반 블록 부호화

3.1 제안 부호화 방법

기존의 임베디드 영상 부호화 방법인 셋분할 부호화 방법 및 블록분할 부호화 방법은 간단한 비트정렬 방식을 사용함으로써 영상의 빠른 압축이 가능한 반면, 비트율-왜곡의 관점에서 최적화되어 있지 않다는 단점이 있다. 반면에 RDE는 임베디드 특징을 유지하면서도 비트율-왜곡 최적화를 수행하는 장점을 가지고 있으나 웨이브렛 계수 단위의 부호화 방법을 사용한다는 단점을 가지고 있다. 또한, EBCOT는 PCRD 최적화를 수행함으로써 기존의 임베디드 영상압축과 차별화되는 우수한 압축 성능을 제공하는 반면 각 코드블록들을 독립적으로 부호화해야 함으로 웨이브렛 계수들간에 존재하는 대역간 상관관계를 이용할 수 없다는 단점이 있다.

본 논문에서 제안하는 "비트율-왜곡 최적화 기반 블록 부호화(Rate-Distortion optimized Block Coding, RDBC)"는 기존의 부호화 방법들의 단점을 개선하기 위해서, 압축 비트스트림의 임베디드 특징을 유지하면서도 웨이브렛 계수나 블록의 비트율-왜곡 최적화를 수행하는 동시에 비트율-왜곡비의 기댓값에 따라 최적화된 블록부호화를 수행한다. 또한 웨이브렛 계수들간에 존재하는 다양한 상관관계를 모델링하여 엔트로피 부호화에 적용한다.

RDBC는 다수의 임베디드 비트스트림을 이용하여 비트율-왜곡 최적화를 수행하는 기존의 PCRD 최적화 방법과는 달리, 웨이브렛 계수나 블록을 부호화함으로서 감소하는 왜곡의 기댓값 E[△D]와 증가하는 비트율의 기댓값 E[△R]을 사용하여 수식 (1)과 같이 정의되는 비트율-왜곡비 기댓값 λ를 구하고, λ에 따라 계수나 블록을 정렬하는 비트율-왜곡 최적화를 수행한다.

$$\lambda = \frac{E[\triangle D]}{E[\triangle R]} \tag{1}$$

수식 (1)의 E[△D]와 E[△R] 값을 구하기 위해서는 웨이브렛 계수의 확률분포, 블록이나 계수의 중요도 확률, 계수의 부호에 대한 확률, 계수의 정제 비트에 대한 확률 정보가 필요하다. RDBC에서는 기존의 연구결과를 바탕으로 웨이브렛 계수의 균등분포(Uniform distribution)를 가정하였으며[13,18], 블록의 중요도 확률은 1/2로[5-8], 계수의 중요도와 부호 및 정제 비트에 대한 확률은 엔트로피 부호화기의 확률 예측기(Probability estimator)[13]를 이용하여 계산한다.

비트율-왜곡비 기댓값 λ에 따라 계수나 블록을 정렬하기 위해서는 일정한 크기로 감소하는 비트율-왜곡 임계값 ϒ가 사용된다. 즉, λ의 값이 ϒ보다 큰 경우에만 해당 블록이나 계수들을 부호화하고 ϒ의 값을 일정한 크기로 감소시키는 일련의 과정을 되풀이함으로서, λ에 따라 블록이나 계수들을 정렬한다. RDBC에서는 λ가 ϒ보다 큰 경우가 발생하면, 웨이브렛 계수의 중요도나 정제 비트 혹은 블록의 중요도를 부호화하는 양자화를 수행한다. 먼저, 계수의 중요도는 수식 (2)와 같이 판단되는 이진 값을 의미하며, 계수의 정제 비트는 크기 임계값에 해당하는 비트평면의 비트 값을 의미한다.

$$S_{T_k}(c_{i,j}) = \begin{cases} 1, & \text{if } T_k \le |c_{i,j}| < T_{k-1} \\ 0, & else \end{cases} \tag{2}$$

수식 (2)에서 T_k, $c_{i,j}$는 각각 크기 임계값 및 (i,j) 좌표에 위치한 계수를 의미하며, $S_{Tk}(c_{i,j})$는 현재의 크기 임계값에 대한 계수 $c_{i,j}$의 중요도를 의미한다. 블록분할 부호화의 경우, RDBC는 기존의 연구들과 차별화되는 새로운 방법을 사용한다. 기존의 연구에서는 블록 B의 중요도를 수식 (3)과 같은 방법으로 판단하고 부호화한 후, 해당 블록이 중요하다고 판단되는 경우 블록분할을 수행한다. 반면, RDBC에서는 블록분할을 기반으로 부호화시에 얻어지는 비트율-왜곡 기댓값 λ_q와 블록에 속한 웨이브렛 계수들을 독립적으로 부호화시에 얻어지는 비트율-왜곡 기댓값 λ_s를 각각 구한 후, 두가지 값 중 큰 값을 λ로 선택하고 이를 Υ와 비교한다. 만약 λ의 값이 Υ보다 크고, λ가 λ_q와 동일한 경우에는 기존의 블록분할 방식과 동일한 부호화 과정을 수행한다. 반면에 λ의 값이 Υ보다 크고, λ가 λ_s와 동일한 경우에는 블록의 중요도 부호화 및 블록분할은 수행되지 않으며, 블록에 속한 모든 계수들은 독립적으로 부호화된다.

$$S_{T_k}(B)=\begin{cases}1, & \text{if } T_k \le \max\limits_{(i,j)\in B}|c_{i,j}| < T_{k-1}\\ 0, & else\end{cases} \qquad (3)$$

엔트로피 부호화를 위해서는 적응적 산술 부호기(Adaptive arithmetic coder)를 사용한다. 적응적 산술 부호기는 컨텍스트를 기반으로 입력 심벌의 부호화 확률을 계산하고 엔트로피 부호화를 수행하는데, RDBC에서는 EBCOT의 인접 계수간 상관관계를 이용하는 컨텍스트에 부모-자식간 상관관계 및 사촌간 상관관계를 추가한 컨텍스트를 사용한다.

3.2 RDBC 부호화 구조

RDBC는 웨이브렛 계수나 블록을 구분하기 위해 LIP(List of Insignificant Pixels), LIB(List of Insignificant Blocks), LSP(List of Significant Pixels)의 세가지 리스트를 사용한다. 이름에서도 알 수 있듯이, 각 웨이브렛 계수나 블록이 가지는 크기 임계값을 기준으로 LIP는 중요하지 않다고 분류된 계수들을 포함하는 리스트이며, LIB는 중요하지 않다고 분류된 블록들을 포함하는 리스트이고, LSP는 중요하다고 판단된 계수들을 포함하는 리스트이다.

그림 4는 RDBC의 부호화 구조를 보여준다. RDBC는 입력되는 웨이브렛 계수에 대한 '초기화(Initialization)', 'LIP 정렬(LIP sorting)', 'LIB 정렬(LIB sorting)', 'LSP 정제(LSP refinement)', '비트율-왜곡 임계값 갱신(R-D threshold update)', '엔트로피 부호화(Entropy coding)'의 6가지 단계를 통해 부호화를 수행함으로서 하나의 임베디드 비트스트림을 생성한다.

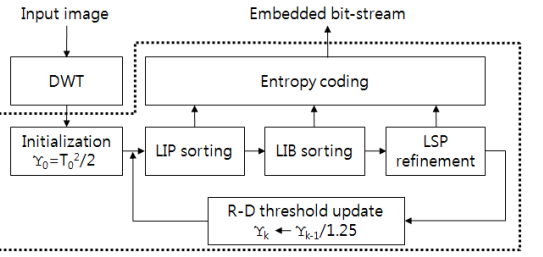


그림 4. RDBC의 부호화 구조
Fig. 4. Encoding structure of RDBC

3.2.1 초기화 단계

초기화 단계에서는 리스트 항목의 중요도를 판단하는 기준인 크기 임계값과 리스트 항목의 부호화 수행 여부를 판단하는 기준인 비트율-왜곡 임계값의 초기값을 결정한다. 먼저, 크기 임계값의 초기값 T_0는 DWT 변환 영상 X 및 (i,j) 좌표에 위치한 웨이브렛 계수 $c_{i,j}$를 이용하여 수식 (4)와 같이 결정된다.

$$T_0 = 2^n, \quad n = \left\lfloor \log_2\left(\max_{(i,j)\in X}|c_{i,j}|\right) \right\rfloor \qquad (4)$$

수식 (4)에서 보듯이, 크기 임계값의 초기값은 웨이브렛 변환 영상에 속한 가장 큰 계수의 첫 번째 비트 평면 n값을 이용하여 2의 승수로 표현된다. 이러한 정보는 부호기 및 복호기에 공통적으로 사용되어야 함으로, n값을 1바이트로 부호화함으로서 T_0의 정보를 복호기에 전달한다. 한편, 비트율-왜곡 임계값의 초기값 Υ_0는 T_0값을 이용하여 수식 (5)와 같이 결정된다.

$$\gamma_0 = \frac{1}{2}T_0^2 \qquad (5)$$

크기 임계값 및 비트율-왜곡 임계값의 초기값이 결정되면 각 리스트에 들어갈 항목을 결정한다. RDBC에서는 저주파 대역인 LL에 속한 계수들을 LIP에 초기화하고 LL을 제외한 부대역들을 블록으로 구성하여 LIB에 초기화한다. LSP는 초기화 단계에서 리스트 항목D LS지 않는다. 또한, LIB에 초기화한다D 리스트에 초기화할 때에는 각 계수나 블록별로 크기 임계값의 초기값을 리스트에 함께 저장한다.

3.2.2 LIP 정렬 단계

LIP 정렬 단계에서는 리스트에 포함된 계수의 중요도를 부호화했을 때 얻어지는 비트율-왜곡비 기댓값을 계산한 후, 계산된 비트율-왜곡비 기댓값에 따라 리스트 계수의 부호화를 수행한다. 리스트 계수의 비트율-왜곡비 기댓값을 구하기 위해서는 리스트 계수를 부호화함으로서 감소하는 왜곡의 기댓값과 증가하는 비트율의 기댓값을 계산한다.

리스트 계수를 부호화함으로서 감소하는 왜곡의 기댓값 $E[\triangle D_p]$는 각 리스트 계수가 가지는 크기 임계값 T_k에 대해 해당 리스트 계수가 수식 (2)를 이용해서 중요하다고 판단될 확률 p_s, 중요하다고 판단된 경우의 복호값 $1.5T_k$를 이용하여 수식 (6)과 같이 계산된다.

$$E[\triangle D_p] = p_s(1.5\,T_k)^2 = p_s 2.25\,T_k^2 \tag{6}$$

리스트 계수를 부호화함으로서 증가하는 비트율의 기댓값 $E[\triangle R_p]$는 각 리스트 계수가 중요하다고 판단될 확률 p_s 및 중요하다고 판단될 경우 부호화되는 부호가 +일 확률 p_+를 이용하여 수식 (7)과 같이 계산된다.

$$E[\triangle R_p] = p_s H(p_+) + H(p_s) \tag{7}$$

여기서 H(p)는 이진 비트가 1일 확률 p에 대한 엔트로피를 나타낸다.

$$H(p) = -p\log_2 p - (1-p)\log_2(1-p) \tag{8}$$

따라서 리스트 계수의 비트율-왜곡비 기댓값 λ_p는 다음과 같이 계산된다.

$$\lambda_p = \frac{p_s 2.25\,T_k^2}{p_s H(p_+) + H(p_s)} = \frac{2.25\,T_k^2}{H(p_+) + H(p_s)/p_s} \tag{9}$$

수식 (9)에 의해 리스트 계수의 비트율-왜곡비 기댓값이 결정되면, 결정된 비트율-왜곡비 기댓값 λ_p와 비트율-왜곡 임계값 Υ_k와의 비교를 통해 리스트 계수의 부호화 여부를 결정한다. 만약 λ_p가 Υ_k보다 작을 경우에는 리스트 계수의 부호화 과정은 생략되며, λ_p가 Υ_k보다 큰 경우에만 리스트 계수의 부호화가 수행된다. 리스트 계수에 대한 부호화는 수식 (2)에 따라 리스트 계수의 중요도를 부호화하는 단계, 리스트 계수의 크기 임계값을 감소시키는 단계, 리스트 계수가 중요하다고 판단된 경우 해당 리스트 계수를 LSP로 이동시키는 단계로 구성된다. 그림 5는 LIP 정렬 과정을 보여준다.

For each entry $c_{i,j}$ in LIP
1. Calculate λ_p using Equation (9)
2. If $\lambda_p > \Upsilon_k$
   2.1. Encode $S_{Tk}(c_{i,j})$
   2.2. Update T_k, $T_{k+1}=T_k/2$
   2.3. If $S_{Tk}(c_{i,j})==1$
      2.3.1. Encode the sign of $c_{i,j}$
      2.3.2. Move the current entry to LSP
3. Else
   3.1. Go to the next entry in LIP

그림 5. LIP 정렬 과정
Fig. 5. LIP sorting procedure

3.2.3 LIB 정렬 단계

LIB 정렬 단계에서는 리스트에 속해있는 블록의 중요도를 부호화했을 때의 비트율-왜곡비 기댓값을 계산한 후, 계산된 비트율-왜곡비 기댓값에 따라 블록의 부호화를 수행한다. 블록의 비트율-왜곡비 기댓값 λ_b는 블록에 속해있는 계수들을 독립적으로 부호화했을 때 얻어지는 비트율-왜곡비 기댓값 λ_s와 그림 6과 같은 블록분할을 기반으로 부호화를 수행했을 때 얻어지는 비트율-왜곡비 기댓값 λ_q를 비교함으로서 다음과 같이 결정된다.

$$\lambda_b = \begin{cases} \lambda_s, & \text{if } \lambda_s \geq \lambda_q \\ \lambda_q, & else \end{cases} \tag{10}$$

λ_s는 블록에 속한 계수들을 독립적으로 부호화했을 때 감소하는 왜곡의 기댓값의 합 $E[\triangle D_s]$와 증가하는 비트율의 기댓값의 합 $E[\triangle R_s]$를 기반으로 수식 (11)과 같이 계산된다.

$$\lambda_s = \frac{E[\triangle D_s]}{E[\triangle R_s]} = \frac{\sum_{(i,j)\in B} E[\triangle D_p]}{\sum_{(i,j)\in B} E[\triangle R_p]} \tag{11}$$

λ_q는 블록분할 부호화를 통해 얻을 수 있는 왜곡 감소 기댓값 $E[\triangle D_q]$와 비트율 증가 기댓값 $E[\triangle R_q]$를 이용하여 계산된다. 블록분할 부호화에서는 수식 (3)과 같은 방법으로 블록의 중요도를 결정하는데, 블록의 중요도가 1인 경우에는 해당 블록 B를 4개의 부블록 $B_i(i\in\{1,2,3,4\})$로 분할하며, 블록의 중요도가 0

인 경우에는 적은 비트를 이용하여 블록내의 전체 비중요 계수를 부호화한다. 따라서 블록의 중요도가 1인 경우에는 4개의 부블록을 반복적으로 부호화함으로서 왜곡의 감소가 발생하며, 0인 경우에는 블록에 속한 전체 비중요 계수들을 적은 비트로 부호화함으로서 비트율 이득이 발생하고 발생된 비트율 이득은 다른 블록이나 계수들을 부호화하는 데에 사용된다. 그러므로 블록의 중요도가 0 또는 1일 확률을 1/2로 가정하고, 블록의 중요도가 0일 경우 얻을 수 있는 비트율 이득을 Ψ로 표시하는 한편 비트율 이득을 이용해서 다른 블록이나 계수들을 부호화함으로서 얻어지는 평균 비트율-왜곡비를 λ_c로 표시하면, $E[\triangle D_q]$는 다음과 같이 계산된다.

$$E[\triangle D_q] = \frac{1}{2}\sum_{i=1}^{4} E[\triangle D_{B_i}] + \frac{1}{2}\Psi\lambda_c \qquad (12)$$

수식 (12)에서 Ψ은 블록 B가 중요하지 않다는 정보를 부호화하기 위한 1비트와 블록에 속한 각 계수들의 중요도가 1일 확률 p_s를 이용하여 다음과 같이 계산된다.

$$\Psi = \sum_{(i,j)\in B} (p_s - 1)\log_2(1-p_s) - 1 \qquad (13)$$

또한 λ_c는 비트율 이득 Ψ를 이용해서 부호화되는 다른 블록이나 계수의 비트율-왜곡비 기댓값이 현재의 비트율-왜곡 임계값 Υ_k보다 크다는 것과 Υ_k는 1.25배의 비율로 감소한다는 것을 고려하여 다음과 같이 계산된다.

$$\lambda_c = \frac{\gamma_k + \gamma_{k-1}}{2} = 1.125\gamma_k \qquad (14)$$

비트율 증가 기댓값 $E[\triangle R_q]$는 블록의 중요도를 부호화하기 위한 1비트와 블록의 중요도가 1인 경우에 부블럭들을 부호화하기 위한 비트율의 합으로 계산된다.

$$E[\triangle R_q] = \frac{1}{2}(1 + \sum_{i=1}^{4} E[\triangle R_{B_i}]) + \frac{1}{2} \qquad (15)$$

따라서 블록분할 부호화를 통해 얻어지는 비트율-왜곡비 기댓값 λ_q는 다음과 같이 계산된다.

$$\lambda_q = \frac{\sum_{i=1}^{4} E[\triangle D_{B_i}] + \Psi\lambda_c}{\sum_{i=1}^{4} E[\triangle R_{B_i}] + 2} \qquad (16)$$

수식 (16)에서 알 수 있듯이, λ_q를 구하기 위해서는 블록분할을 반복적으로 수행하며 $E[\triangle D_q]$와 $E[\triangle R_q]$을 계산해야 한다. RDBC에서는 λ_q를 효율적으로 계산하기 위해서 영상 분할 영역에서 널리 사용되고 있는 분할-합병(Split-merge)[19]의 개념을 이용한다. 먼저 분할 과정에서는 입력되는 블록에 그림 6과 같은 블록분할 과정을 반복적으로 수행하여 쿼드트리(Quad-tree)를 생성한다. 다음으로 합병 과정에서는 트리노드의 종단에서부터 상단 방향으로 트리노드 합병을 수행하는데, 이때 수식 (11), (12), (15), (16)을 기반으로 합병된 트리 노드 υ의 $E[\triangle D_{s,v}]$, $E[\triangle R_{s,v}]$, $E[\triangle D_{q,v}]$, $E[\triangle R_{q,v}]$, $\lambda_{s,\upsilon}$ 및 $\lambda_{q,\upsilon}$를 계산한다. 이러한 방법으로 얻어진 값들은 합병된 트리 노드의 왜곡 감소 기댓값 $E[\triangle D_\upsilon]$와 증가 비트율 기댓값 $E[\triangle R_\upsilon]$ 계산에 사용되며, LIB에서는 $\lambda_q > \lambda_s$ 및 $\lambda_q > \Upsilon_k$ 조건을 만족할 때에만 블록분할 부호화가 수행됨으로, 아래와 같은 조건에 따라 합병된 트리 노드의 $E[\triangle D_\upsilon]$ 및 $E[\triangle R_\upsilon]$를 계산한다.

1) if $\lambda_{q,\upsilon} > \lambda_{s,\upsilon}$ and $\lambda_{q,\upsilon} > \Upsilon_k$, then $E[\triangle D_\upsilon] = E[\triangle D_{q,v}]$, $E[\triangle R_\upsilon] = E[\triangle R_{q,v}]$

2) else $E[\triangle D_\upsilon] = E[\triangle D_{s,v}]$, $E[\triangle R_\upsilon] = E[\triangle R_{s,v}]$

이와 같은 방법으로 트리 합병을 반복적으로 수행함으로서, 최종 트리 노드의 $E[\triangle D_\upsilon]$ 및 $E[\triangle R_\upsilon]$를 구할 수 있으며, 이 값들은 각각 수식 $E[\triangle D_q]$ 및 $E[\triangle R_q]$에 해당됨으로 수식 (16)에 따라 λ_q가 계산된다.

λ_q 및 λ_s의 값이 결정되면 수식 (10)에 따라 λ_b를 얻을 수 있으며, λ_b와 비트율-왜곡 임계값 Υ_k의 비교를 통해 부호화 여부가 결정된다. RDBC에서는 λ_b의 값이 Υ_k보다 큰 경우에만 블록 부호화를 수행하는데 수식 (10)에 따라 λ_b가 λ_q로 결정된 경우에는 블록의 중요도 부호화, 블록의 크기 임계값 갱신 과정이 수행되며, 추가적으로 블록의 중요도가 1인 경우에는 블록분할, 부블럭의 LIB 리스트 추가, 현재 블록 삭제와 같은 일련의 과정이 반복된다. 반면에 λ_b의 값이 Υ_k보다 크다는 조건이 만족하고 수식(10)에 따라 λ_b가 λ_s로 결정된 경우에는 블록내의 모든 계수들에 대한 독

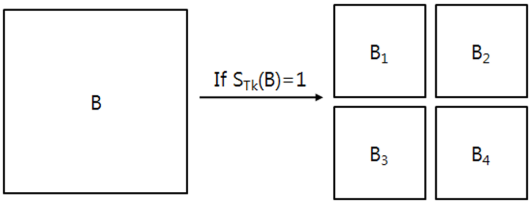


그림 6. 블록분할
Fig. 6. Block partitioning

For each entry B in LIB
1. Calculate λ_b using λ_s, λ_q, and Equation (10)
2. If $\lambda_b > \Upsilon_k$
  2.1. If($\lambda_b == \lambda_q$)
    2.1.1. Encode $S_{Tk}(B)$
    2.1.2. Update T_k, $T_{k+1}=T_k/2$
    2.1.3. If $S_{Tk}(B)==1$
      2.1.3.1. Add 4 sub-blocks to LIB with T_{k+1}
      2.1.3.2. Remove B from LIB
  2.2. Else
    2.2.1. For each coefficients $c_{i,j}$ in the block
      2.2.1.1.Calculate λ_p using Equation (9)
      2.2.1.2. If $\lambda_p > \Upsilon_k$
        - Encode $S_{Tk}(c_{i,j})$
        - Update T_k, $T_{k+1}=T_k/2$
        - If $S_{Tk}(c_{i,j})==1$
          * Encode the sign of $c_{i,j}$
          * Add $c_{i,j}$ to LSP
        - Else
          * Add $c_{i,j}$ to LIP
      2.2.1.3. Else
        - Add $c_{i,j}$ to LIP
    2.2.2. Remove B from LIB
3. Else
  3.1. Go to the next entry in LIB

그림 7. LIB 정렬 과정
Fig. 7. LIB sorting procedure

립적인 부호화가 수행된다. 그림 7은 LIB 정렬 과정을 보여준다.

3.2.4 LSP 정제 단계

LSP 정제 단계는 LIP 및 LIB 정렬 단계를 통해 중요하다고 판단된 계수들에 대한 정제 비트를 부호화하는 단계로서, 정제 부호화를 통해 얻어지는 비트율-왜곡비가 비트율-왜곡 임계값보다 큰 경우에만 부호화를 수행한다. LSP 리스트 계수들을 부호화함으로서 얻어지는 비트율-왜곡비 기댓값 λ_r은 정제 부호화를 통해 감소하는 왜곡의 기댓값 $E[\triangle D_r]$과 증가하는 비트율의 기댓값 $E[\triangle R_r]$을 이용해서 계산된다. 리스트 계수에 정제 부호화가 수행되면 정제화 비트 값에 관계없이 복호화 오류는 $0.5T_k$만큼 줄어들게 됨으로, 정제 비트가 1일 확률을 p_r이라고 할 때 $E[\triangle D_r]$과 $E[\triangle R_r]$는 각각 다음과 같이 계산된다.

$$E[\triangle D_r] = p_r(0.5T_k)^2 + (1-p_r)(0.5T_k)^2 = 0.25T_k^2 \quad (17)$$

For each entry $c_{i,j}$ in LSP
1. Calculate λ_r using Equation (19)
2. If $\lambda_r > \Upsilon_k$
  2.1. Encode n-th bit of $|c_{i,j}|$, $n=\log_2 T_k$
  2.2. Update T_k, $T_{k+1}=T_k/2$
3. Else
  3.1. Go to the next entry in LSP

그림 8. LSP 정제 과정
Fig. 8. LSP refinement procedure

$$E[\triangle R_r] = H(p_r) \quad (18)$$

따라서 λ_r은 다음과 같이 계산된다.

$$\lambda_r = \frac{0.25T_k^2}{H(p_r)} \quad (19)$$

수식 (19)에 의해서 λ_r이 결정되면, λ_r과 비트율-왜곡 임계값 Υ_k와의 비교를 통해 λ_r의 값이 Υ_k보다 클 경우에 정제 부호화를 수행한다. 정제 부호화에서는 리스트 계수 $c_{i,j}$의 크기 임계값 T_k를 이용하여, $|c_{i,j}|$의 $\log_2 T_k$번째 비트가 부호화된다. 그림 8은 LSP 정제 과정을 보여준다.

3.2.5 비트율-왜곡 임계값 갱신 단계

비트율-왜곡 임계값 갱신 단계에서는 각 리스트 항목의 부호화 여부 판단기준인 비트율-왜곡 임계값 Υ_k를 수식 (20)과 같이 갱신한 후, LIP 정렬 부호화 단계로 되돌아가는 과정을 수행한다.

$$\gamma_k = \gamma_{k-1}/1.25 \quad (20)$$

3.2.6 엔트로피 부호화 단계

RDBC는 계수들의 중요도, 계수들의 부호, 블록의 중요도, 정제 비트를 엔트로피 부호화하기 위해 적응적 산술 부호기를 사용한다. 웨이브렛 계수의 중요도를 엔트로피 부호화하기 위해서는 인접한 8개의 계수와 부모 및 사촌 노드에 해당하는 3개 계수의 중요도 정보를 사용한다. 그러나 11개 계수의 중요도 정보는 211개의 컨텍스트를 생성함으로, 컨텍스트 희석화(Context dilution)[20] 문제가 발생할 수 있다. 이러한 문제점을 해결하고자, RDBC는 인접한 8개의 웨이브렛 계수 정보를 기반으로 28개의 컨텍스트를 9개의 컨텍스트로 양자화 하는 EBCOT의 컨텍스트에 부모 및 사촌 노드의 중요 계수의 숫자를 조합함으로서 27개의 컨텍스트를 정의하고 이를 엔트로피 부호화에

사용한다. 또한, 계수들의 부호 및 정제 비트의 엔트로피 부호화를 위해서는 EBCOT와 동일한 컨텍스트를 사용하였으며, 블록의 중요도는 1비트를 사용하여 부호화한다.

Ⅳ. 실험 결과

DWT에 사용되는 웨이브렛 필터의 선택은 영상 압축 성능을 좌우하는 중요한 요소 중 하나이다. 표 1은 본 논문에서 제안한 RDBC에 다양한 웨이브렛 필터를 적용하여 얻은 영상 압축 성능을 보여준다. 실험을 위해서는 모든 입력 영상에 5레벨 Dyadic DWT가 적용되었으며, DWT를 위한 웨이브렛 필터로는 영상 압축에 좋은 성능을 제공하는 것으로 알려진 Daub9/7, Daub6, Daub8, Villa18/10, Bior5.5 및 Bior6.8이 사용되었다. 또한 실험 영상으로는 압축 영역에서 가장 많이 사용되는 512×512 크기의 Lenna, Barbara, Goldhill과 JPEG2000 표준의 실험 영상인 2560×2048 크기의 Bike, Cafe, Woman이 사용되었다. 복원된 영상의 성능을 측정하는 방법으로는 원 영상과 복원된 영상의 MSE(Mean Squared Error)를 기반으로 수식 (21)과 같이 정의되는 PSNR(Peak Signal-to-Noise Ratio)을 사용하였다.

$$PSNR = 10\log_2(\frac{255^2}{MSE})\,dB \tag{21}$$

표 1에서 보듯이, RDBC에서는 고주파 영역이 많은 일부 영상의 경우에 Villa18/10 필터가 좋은 성능을 제공하지만, 다양한 영상에 대한 평균적인 성능은 Bior6.8 및 Daub9/7 필터를 사용하였을 때 가장 우수하다.

표 2는 본 논문에서 제안한 RDBC에 Daub9/7 (D9/7)과 Bior6.8(B6.8) 필터를 적용하였을 때의 결과와 기존의 대표적인 임베디드 부호화 방법인 SPIHT(Set Partitioning in Hierarchical Trees)[6] 및 EBCOT[14]의 성능을 비교한 것이다. RDBC와 마찬가지로 SPIHT 및 EBCOT는 5 레벨 Dyadic DWT를 사용하며, DWT를 위해서는 Daub9/7 필터를 사용한다. 표 2에 표시한 EBCOT의 성능은 임베디드 비트스트림을 생성하는 Generic 부호화 모드의 결과로서, 부호화시 64×64 크기의 코드블록과 블록분할 부호화를 이용하는 한편 부블럭의 크기가 16×16이 되면 더 이상 블록분할 부호화를 수행하지 않는 방식을 사용한다. 표 2에서 보듯이, RDBC는 Daub9/7 및 Bior6.8 필터

표 1. 웨이브렛 필터에 따른 RDBC의 PSNR 비교
Table 1. PSNR comparisons of RDBC according to wavelet filters

Lenna(512×512)							Barbara(512×512)						
Bit Rate	Daub 9/7	Daub6	Daub8	Villa 18/10	Bior 5.5	Bior 6.8	Bit Rate	Daub 9/7	Daub6	Daub8	Villa 18/10	Bior 5.5	Bior 6.8
0.0625	28.6	27.56	27.79	28.59	28.27	**28.62**	0.0625	23.5	23.11	23.17	23.53	23.36	**23.54**
0.125	31.43	30.39	30.58	31.49	31.15	**31.51**	0.125	25.28	24.76	24.88	25.46	25.14	**25.51**
0.25	34.48	33.5	33.7	34.51	34.07	**34.6**	0.25	28.24	27.56	27.8	28.5	28	**28.57**
0.5	37.6	36.81	37.02	37.6	37.16	**37.67**	0.5	32.11	31.13	31.54	32.47	31.95	**32.56**
1.0	40.65	40.19	40.32	40.63	40.32	**40.71**	1.0	37.14	36.33	36.71	**37.56**	36.75	**37.56**
Goldhill(512×512)							Bike(2560×2048)						
Bit Rate	Daub 9/7	Daub6	Daub8	Villa 18/10	Bior 5.5	Bior 6.8	Bit Rate	Daub 9/7	Daub6	Daub8	Villa 18/10	Bior 5.5	Bior 6.8
0.0625	26.87	26.2	26.13	26.88	26.73	**26.92**	0.0625	23.61	23.23	23.27	**23.64**	23.55	23.58
0.125	28.69	28.16	28.16	28.65	28.29	**28.71**	0.125	**26.36**	25.64	25.79	26.08	26.05	26.29
0.25	**30.79**	30.35	30.35	30.67	30.56	30.77	0.25	29.59	29.16	29.19	29.45	29.15	**29.65**
0.5	33.41	32.98	32.98	33.37	33.06	**33.45**	0.5	**33.6**	33.13	33.14	33.39	33.09	33.58
1.0	36.76	36.5	36.5	36.85	36.11	**36.87**	1.0	**38.16**	37.83	37.82	38.06	37.64	**38.16**
Cafe(2560×2048)							Woman(2560×2048)						
Bit Rate	Daub 9/7	Daub6	Daub8	Villa 18/10	Bior 5.5	Bior 6.8	Bit Rate	Daub 9/7	Daub6	Daub8	Villa 18/10	Bior 5.5	Bior 6.8
0.0625	**19.14**	18.97	19	19.08	18.95	**19.14**	0.0625	25.54	25.36	25.42	25.52	25.34	**25.62**
0.125	20.88	20.63	20.65	20.83	20.74	**20.9**	0.125	27.42	27.12	27.23	27.5	27.26	**27.52**
0.25	**23.23**	23.07	23.1	23.17	23.15	23.19	0.25	30.13	29.69	29.84	30.14	29.88	**30.21**
0.5	27.06	26.69	26.73	26.87	26.57	**27.08**	0.5	33.86	33.42	33.56	33.92	33.48	**33.96**
1.0	**32.43**	32.02	32.03	32.35	31.89	**32.43**	1.0	38.71	38.27	38.39	38.55	38.44	**38.72**

표 2. RDBC와 다른 임베디드 압축 방법과의 PSNR 비교
Table 2. PSNR comparisons between RDBC and other embedded image compression schemes

Lenna(512×512)							Barbara(512×512)						
Bit Rate	SPIHT	EBCOT	RDBC(D 9/7)	RDBC(B 6.8)	Gain 1	Gain 2	Bit Rate	SPIHT	EBCOT	RDBC (D9/7)	RDBC (B6.8)	Gain 1	Gain 2
0.0625	28.38	28.1	28.6	**28.62**	0.5	0.52	0.0625	23.35	23.34	23.5	**23.54**	0.16	0.2
0.125	31.10	31.05	31.43	**31.51**	0.38	0.46	0.125	24.86	25.37	25.28	**25.51**	-0.09	0.14
0.25	34.14	34.16	34.48	**34.6**	0.32	0.44	0.25	27.58	28.4	28.24	**28.57**	-0.16	0.17
0.5	37.25	37.29	37.6	**37.67**	0.31	0.38	0.5	31.40	32.29	32.11	**32.56**	-0.18	0.27
1.0	40.45	40.48	40.65	**40.71**	0.17	0.23	1.0	36.41	37.11	37.14	**37.56**	0.03	0.45
Sum	-	-	-	-	**1.68**	**2.03**	Sum	-	-	-	-	**-0.24**	**1.23**
Goldhill(512×512)							Bike(2560×2048)						
Bit Rate	SPIHT	EBCOT	RDBC (D9/7)	RDBC (B6.8)	Gain 1	Gain 2	Bit Rate	SPIHT	EBCOT	RDBC (D9/7)	RDBC (B6.8)	Gain 1	Gain 2
0.0625	26.73	26.6	26.87	**26.92**	0.27	0.32	0.0625	23.44	**23.78**	23.61	23.58	-0.17	-0.2
0.125	28.48	28.51	28.69	**28.71**	0.18	0.2	0.125	25.89	**26.37**	26.36	26.29	-0.01	-0.08
0.25	30.56	30.59	**30.79**	30.77	0.2	0.18	0.25	29.12	29.6	29.59	**29.65**	-0.01	0.05
0.5	33.13	33.25	33.41	**33.45**	0.16	0.2	0.5	33.01	33.46	**33.6**	33.58	0.14	0.12
1.0	36.55	36.59	36.76	**36.87**	0.17	0.28	1.0	37.70	38.09	**38.16**	**38.16**	0.07	0.07
Sum	-	-	-	-	**0.98**	**1.18**	Sum	-	-	-	-	**0.02**	**-0.04**
Cafe(2560×2048)							Woman(2560×2048)						
Bit Rate	SPIHT	EBCOT	RDBC (D9/7)	RDBC (B6.8)	Gain 1	Gain 2	Bit Rate	SPIHT	EBCOT	RDBC (D9/7)	RDBC (B6.8)	Gain 1	Gain 2
0.0625	18.95	19.06	**19.14**	**19.14**	0.08	0.08	0.0625	25.43	**25.63**	25.54	25.62	-0.09	-0.01
0.125	20.67	20.82	20.88	**20.9**	0.06	0.08	0.125	27.33	27.39	27.42	**27.52**	0.03	0.13
0.25	23.03	23.2	**23.23**	23.19	0.03	-0.01	0.25	29.95	30.04	30.13	**30.21**	0.09	0.17
0.5	26.49	26.87	27.06	**27.08**	0.19	0.21	0.5	33.59	33.7	33.86	**33.96**	0.16	0.26
1.0	31.74	32.03	**32.43**	**32.43**	0.4	0.4	1.0	38.28	38.49	38.71	**38.72**	0.22	0.23
Sum	-	-	-	-	**0.76**	**0.76**	Sum	-	-	-	-	**0.41**	**0.78**

를 사용하였을 때 실험영상의 전체 비트율에서 SPIHT과 비교하여 우수한 성능을 제공하는데, Daub9/7 필터를 사용하는 경우에는 약 0.11~0.73dB의 성능향상을 보이며, Bior6.8 필터를 사용하는 경우에는 약 0.14~1.16dB의 성능향상을 보인다.

표 2의 다섯 번째 및 여섯 번째 칼럼인 Gain 1과 Gain 2는 각각 RDBC에 Daub9/7 필터와 Bior6.8 필터를 적용하였을 때의 성능과 EBCOT의 성능을 비교하여 얻을 수 있는 PSNR 이득을 보여준다. 표 2에서 확인할 수 있듯이, RDBC는 EBCOT와 비교하여 전체적으로 우수한 성능을 제공한다. 특히, Lenna나 Goldhill과 같이 영상내의 고주파 성분이 적은 평활한 영상의 경우에는 모든 비트율에서 우수한 성능을 제공한다. 반면에 Barbara나 Bike와 같이 고주파 성분이 많은 영상의 경우에는 EBCOT와 비슷한 성능을 제공한다. 이러한 현상은 EBCOT 및 RDBC의 부호화 방법 차이에서 기인한다. EBCOT는 블록분할 부호화를 기반으로 코드블록들을 독립적으로 부호화하여 다수의 임베디드 비트스트림을 생성한 후, 각 비트스트림을 생성하기 위해 실제로 사용된 비트율 및 왜곡의 정보를 기반으로 PCRD 최적화를 수행한다. 따라서 블록분할 부호화를 통해 많은 이득을 얻을 수 있는 평활한 영상의 경우보다는 블록분할 부호화를 통해 많은 이득을 얻을 수 없는 즉, 고주파 성분이 많은 영상에 대해서 보다 많은 장점을 가진다. 반면에 RDBC는 비트율-왜곡비 기댓값에 따라 계수나 블록을 정렬함으로서 비트율-왜곡 최적화를 수행하는 동시에 블록분할 부호화를 수행한다. 따라서 고주파 성분이 많은 영상 보다는 평활한 영상의 경우에 비트율-왜곡비 기댓값이 실제의 비트율-왜곡비와 비교하여 정확도가 높아짐으로, 평활한 영상의 압축에 보다 많은 장점을 가진다.

표 2를 통해 비트율이 높아짐에 따라 RDBC의 부호화 이득이 EBCOT와 비교하여 평균적으로 증가함을 알 수 있다. 이러한 이유는 엔트로피 부호화를 위한 컨텍스트 모델에 기인한 것으로 분석된다. 즉, EBCOT는 각 코드블록들을 독립적으로 부호화해야 함으로 엔트로피 부호화를 위한 확률 통계적 샘플이

RDBC와 비교하여 상대적으로 적을 뿐만 아니라 웨이브렛 계수의 대역간 상관관계를 이용할 수 없는 반면, RDBC는 엔트로피 부호화를 위해 상대적으로 많은 확률 통계적 샘플을 이용할 수 있으며, 웨이브렛 계수의 대역간 상관관계도 이용할 수 있기 때문인 것으로 분석된다.

Ⅴ. 결 론

본 논문에서는 압축 비트스트림의 임베디드 특징을 유지하면서도 기존 부호화 방법들의 단점을 개선한 새로운 부호화 방법인 RDBC를 제안하였다. 제안된 RDBC는 웨이브렛 계수나 블록을 비트율-왜곡비 기댓값에 따라 정렬함으로서 비트율-왜곡 최적화를 수행하는 동시에 비트율-왜곡비 기댓값에 따라 최적화된 블록분할 부호화를 수행한다. 또한 RDBC는 웨이브렛 계수간에 존재하는 다양한 상관관계를 엔트로피 부호화에 적용한다. 실험 결과는 RDBC가 기존의 대표적인 임베디드 부호화 방법인 SPIHT 및 EBCOT와 비교하여 향상된 압축 성능을 제공함을 보여준다. RDBC는 SPIHT과 비교하여 다양한 비트율에서 0.11~1.16dB의 PSNR 성능 향상을 제공하며, EBCOT와 비교하여 -0.18~0.52dB의 PSNR 성능 향상을 제공한다. 특히, RDBC는 고주파 성분이 적은 영상일수록 혹은 비트율이 높아질수록 기존의 부호화 방법과 비교하여 우수한 성능을 제공한다.

본 논문에서 제안한 RDBC는 웨이브렛 계수나 블록의 비트율-왜곡비 기댓값을 계산해야 함으로 EBCOT와 같은 비트율-왜곡 최적화를 수행하는 부호화 방법과는 유사한 복잡도를 가지지만 SPIHT과 같이 웨이브렛 계수의 크기에 따라 셋 혹은 블록 분할 부호화를 수행하는 방법보다는 많은 연산량을 가진다. 향후의 연구 과제로는 RDBC의 연산량을 감소시키는 방법을 개발하는 것과 RDBC의 개념을 확장하여 컬러 영상 부호화에 적용하는 것이다.

References

[1] N. Kim, T. Song, W. Kim, and S. Pack, “A scalable video coding (SVC)-aware retransmission scheme for multimedia streaming in IEEE 802.11 WLANs,” *J. KICS*, vol. 39, no. 2, pp. 95-101, Feb. 2014.

[2] J. Wang, J. Park, and Y. Kim, “Layer selective cooperation using superposition coding for reduction of expected distortion,” *J. KICS*, vol. 37, no. 7, pp. 517-527, Jul. 2012.

[3] J. Noh and M. Hong, “Fast integer-pel motion estimation based on statistical property for H.264/AVC,” *J. KICS*, vol. 37, no. 8, pp. 669-678, Aug. 2012.

[4] A. A. Moinuddin, E. Khan, and M. Ghanbari, “Efficient algorithm for very low bit rate embedded image coding,” *IET Image Process.*, vol. 2. no. 2, pp. 59-71, Apr. 2008.

[5] J. M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3445-3462, Dec. 1993.

[6] A. Said and W. A. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Trans. Circuits and Syst. for Video Tech.*, vol. 6, no. 3, pp. 243-250, Jun. 1996.

[7] H. Zhu, C. Xiu, and D. Yang, “An improved SPIHT algorithm based on wavelet coefficient blocks for image coding,” *Int. Conf. Comput. Appl. Syst. Modeling*, pp. 646-649, Taiyuan, China, Oct. 2010.

[8] W. A. Pearlman and A. Said, “Set partition coding: part I of set partition coding and image wavelet coding systems,” *Foundations and Trends in Signal Process.*, vol. 2, no. 2, pp. 95-180, Nov. 2008.

[9] A. Islam and W. A. Pearlman, “An embedded and efficient low-complexity hierarchical image coder,” *Visual Comm. Image Process.*, pp. 294-305, San Jose, USA, Jan. 1999.

[10] W. A. Pearlman, A. Islam, N. Nagaraj, and A. Said, “Efficient, low-complexity image coding with a set-partitioning embedded block coder,” *IEEE Trans. Circuits and Syst. for Video Tech.*, vol. 14, no. 11, pp. 1219-1235, Nov. 2004.

[11] R. Ngadiran, S. Boussakta, A. Bouridane, and F. Khelifi, “Exploiting chrominance planes similarity on listless quadtree coders,” *IET Image Process.*, vol. 7, no. 6, pp. 575-585, Aug. 2013.

[12] Z. Xiong, K. Ramchandran, and M. T.

Orchard, "Space-frequency quantization for wavelet image coding," *IEEE Trans. Image Process.*, vol. 6, no. 5, pp. 677-693, May 1997.

[13] J. Li and S. Lei, "An embedded still image coder with rate-distortion optimization," *IEEE Trans. Image Process.*, vol. 8, no. 7, pp. 913-924, Jul. 1999.

[14] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Process.*, vol. 9, no. 7, pp. 1158-1170, Jul. 2000.

[15] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 7, pp. 674-693, Jul. 1989.

[16] M. Omidyeganeh, S. Ghaemmaghami, and S. Shirmohammadi, "Application of 3D-wavelet statistics to video analysis," *Multimedia Tools Appl.*, vol. 65, no. 3, pp 441-465, Aug. 2013.

[17] R. W. Buccigrossi and E. P. Simoncelli, "Image compression via joint statistical characterization in the wavelet domain," *IEEE Trans. Image Process.*, vol. 8, no. 12, pp 1688-1701, Dec. 1999.

[18] F. Auli-Llinas, "General embedded quantization for wavelet-based lossy image coding," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1561-1574, Mar. 2013.

[19] M. Silveira, J. C. Nascimento, J. S. Marques, A. R. S. Marcal, T. Mendonca, S. Yamauchi, J. Maeda, and J. Rozeira, "Comparison of segmentation methods for melanoma diagnosis in dermoscopy images," *IEEE J. Sel. Topics Signal Process.*, vol. 3, no. 1, pp. 35-45, Feb. 2009.

[20] S. Forchhammer and X. Wu, "Context quantization by minimum adaptive code length," *IEEE Int. Symp. Inf. Theory*, pp. 246-250, Nice, France, Jun. 2007.

양 창 모 (Chang Mo Yang)

1998년 2월 : 한국항공대학교 항공전자공학과 졸업
2000년 2월 : 광주과학기술원 정보통신공학과 석사
2002년 2월 : 광주과학기술원 정보통신공학과 박사수료
2002년 3월~현재 : 전자부품연구원 책임연구원
2012년 9월~현재 : 광운대학교 전자통신공학과 박사과정
<관심분야> 영상 압축, 실시간 비디오 스트리밍

정 광 수 (Kwangsue Chung)

1981년 2월 : 한양대학교 전자공학과 졸업
1983년 2월 : 한국과학기술원 전기 및 전자공학과 석사
1991년 2월 : University of Florida 전기공학과 박사
1983년 3월~1993년 2월 : 한국전자통신연구원 선임연구원
1993년 3월~현재 : 광운대학교 전자통신공학과 교수
<관심분야> 인터넷 QoS, 유·무선 비디오 스트리밍, 센서 네트워크