

# EBCOT 기반의 적응적 관심영역 코딩

강기준<sup>†</sup>, 이부권<sup>\*\*</sup>, 서영건<sup>\*\*\*</sup>

## 요 약

하나의 영상에서 원하는 일부분만 고화질로 압축을 하거나, 우선적으로 전송하기 위하여 JPEG2000 표준에서 관심영역(ROI) 코딩 방법을 제공한다. ROI 코딩에서 중요한 것은 ROI와 배경간의 상대적인 중요도 처리와 ROI 마스크 처리이다. 본 연구에서는 ROI 마스크 정보를 줄이고 화질을 개선하기 위하여, 기존의 묵시적 ROI 방법과 수정된 묵시적 ROI 코딩 방법을 보완한 적응적 ROI 코딩 방법을 제안한다. 제안한 방법은 압축된 비트 스트림으로부터 ROI를 추출하는 EBCOT 기반의 ROI 코딩 방법이며, 코드블록을 6가지 패턴으로 분류하여 ROI 마스크 정보를 구한다. 마스크 정보에는 코드블록 당 패턴 정보(3비트)와 두 영역간의 경계의 폭을 나타내는 정보(5비트)가 포함된다. 실험 결과, ROI에서 뿐만 아니라 전체 영역에서도 압축 성능이 우수함을 보였다.

## An Adaptive Region-of-Interest Coding Based on EBCOT

Ki-Jun Kang<sup>†</sup>, Bu-Kwon Lee<sup>\*\*</sup>, Yeong-Geon Seo<sup>\*\*\*</sup>

## ABSTRACT

To compress a specific part of an image with high quality or to transfer it, JPEG2000 standard offers an ROI(Region-of-Interest) image coding method. What is important in ROI coding is to process relative importance between ROI and background and to process ROI mask. We propose an adaptive ROI coding method supplemented the existing Implicit ROI coding and Modified implicit ROI coding to improve image quality and reduce ROI mask information. The proposed method is an EBCOT-based ROI coding that extracts ROI from the compressed bitstream, and gets the ROI mask information by classifying the codeblocks into 6 patterns. The information includes the pattern type(3bit) and the width(5bit) expressing the boundary between two regions for each codeblock. As a result, the method shows an excellent compression performance in ROI region as well as in the whole region of an image.

**Key words:** JPEG2000, EBCOT, ROI(관심영역)

## 1. 서 론

인간이 받아들이는 정보 중에 영상은 전체 정보 중 상당한 부분을 차지할 뿐만 아니라 단순한 텍스트보다 정보 전달 수단으로 효과가 뛰어나, 의료진단, 웹 검색, 영상 데이터베이스 그리고 컴퓨터 통신 등 다양한 응용 분야에서 활용되고 있다. 하지만 영상은

고 용량으로 인해, 저장과 전송을 위해 많은 메모리와 넓은 대역폭을 요구하게 되어, 사용자의 다양한 요구를 만족시키기 어렵다. 다행히 영상은 큰 중복성을 갖고 있어, 원 영상의 질을 시각적으로 크게 저하시키지 않는 범위 내에서 압축하는 기술에 대한 표준화가 여러 표준화 기관들을 중심으로 활발히 추진되고 있다[1].

※ 교신저자(Corresponding Author) : 서영건, 주소 : 경남 진주시 900번지(660-701), 전화 : 055)751-6182, FAX : 055)751-6117 E-mail : young@gsnu.ac.kr

접수일 : 2006년 6월 2일, 완료일 : 2006년 9월 18일

<sup>†</sup> 준회원, 경상대학교 대학원 컴퓨터과학과

(E-mail : kijuny@cmpal.com)

<sup>\*\*</sup> 정회원, 경상대학교 컴퓨터과학과, 정보통신연구소원  
(E-mail: bklce@gsnu.ac.kr)

<sup>\*\*\*</sup> 종신회원, 경상대학교 컴퓨터교육과, 정보통신연구소원

영상 압축 표준의 대표인 JPEG2000(Joint Photographic Experts Group 2000)은 기존 영상 압축 표준에서 제공하지 않는 새로운 기능들을 제공할 뿐만 아니라 더욱 향상된 압축 성능을 보인다. 주요 특징 중에 하나인 ROI 코딩은 영상 내의 ROI(Region-Of-Interest)를 배경보다 ROI 영상을 더 나은 품질로 우선 처리하는 기술로서 영상 내의 특정 영역이 다른 영역보다 높은 중요성을 가지는 응용에 적용할 수 있다[1-3].

대부분의 응용에는 인코딩 시에 ROI를 결정하는 정적 ROI 방법을 사용하지만, 압축된 비트 스트림으로부터 임의의 ROI를 추출하는 응용에는 디코딩 시에 ROI 정보를 제공받아 코딩하는 동적 ROI 방법을 적용해야 한다[4]. 동적 ROI 방법은 EBCOT(Embedded Block Coding with Optimized Truncation algorithm) 기반의 목시적 방법[5], 수정된 목시적 방법[6], Fast ROI Transcoding[4], Flexible and dynamic[7], Prioritized[8] 등이 있다.

목시적 방법은 코드블록 내에 ROI WC(Wavelet Coefficient)가 포함하면 ROI 코드블록으로 처리함으로써 ROI 코드블록 내의 배경 WC도 우선처리를 하는 문제가 발생한다. 이 문제를 보완한 수정된 목시적 방법은 ROI 코드블록 내의 배경 WC를 삭제하여 우선권을 줄인다. 이 방법은 ROI 코드블록 내의 배경 WC의 우선처리 문제는 개선되었지만, 삭제로 인해 손실 문제가 발생한다.

본 논문에서는 EBCOT 기반의 적응적 ROI 코딩 방법을 제안한다. 제안된 방법은 ROI 코드블록 내의 배경 WC의 양과 분포된 모양에 따라 빠르게 패턴 기반의 적응적인 ROI 코드블록을 식별하고, 디코더에서 ROI 마스크 생성과 ROI 코드블록을 식별하여

적은 양의 ROI 코드블록 식별정보를 인코더로 전송하여 코딩 성능을 높인다. 본 논문의 구성은 2장에서 관련연구, 3장에서 제안한 기법을 설명하고 4장에서 실험결과를 보인다.

## 2. 관련연구

### 2.1 EBCOT

JPEG2000은 EBCOT 알고리즘을 이용한 웨이블릿 기반의 영상 압축 표준으로써 인코딩 과정은 영상을 다수 해상도 레벨, 서브밴드(Subband), 프리신트(Precinct), 코드블록을 포함하는 계층적 조직으로 분해한다. 인코더는 우선 RGB로 나누어진 영상의 색상 성분을 YCbCr 색상 성분으로 변환한다. 이어서, 이산 웨이블릿 변환(DWT : Discrete Wavelet Transform)을 수행한다. 변환된 영상의 각 성분은 몇몇의 해상도 레벨들로 나누어진다. 각 해상도 레벨은 서브밴드의 열로 구성되고, 가장 낮은 해상도 레벨은 네 개의 서브밴드를 갖고, 그 외 레벨은 세 개의 서브밴드를 갖는다. 각 서브밴드에 있는 WC는 양자화된 다음, EBCOT 코딩을 수행하기 위해서 규칙적인 프리신트와 코드블록의 배열로 나누어진다.

EBCOT 코딩 과정은 Tier-1과 Tier-2로 구성되어 있다[그림 2]. Tier-1 코딩은 코드블록을 하나씩 독립적으로 인코딩한다. 코드블록의 양자화된 WC는 부호 비트와 절대 값으로 나누어진다. 각각의 절대 값은 크기가 같은 비트들을 모아 비트-평면(Bit-Plane)을 생성하게 된다. 각 코드블록은 3 가지의 코딩 패스들을 이용하여 상위 비트-평면(MSB : Most Significant Bit-Plane)부터 하위 비트-평면

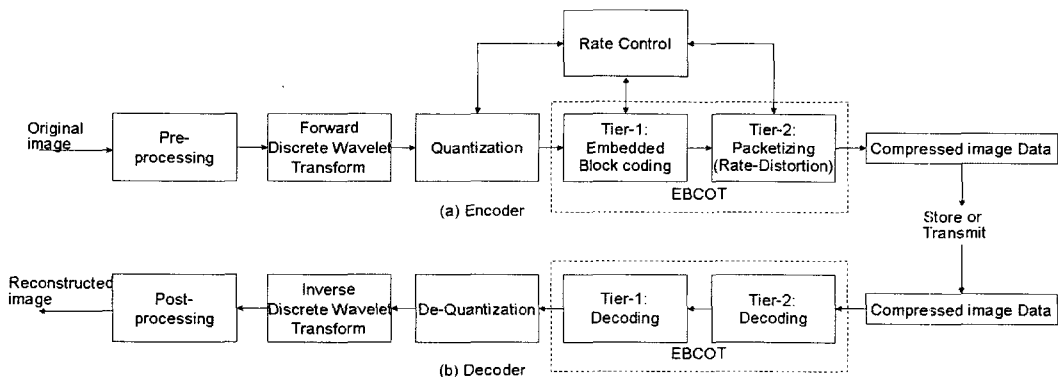


그림 1. JPEG2000 알고리즘의 기본 블록도

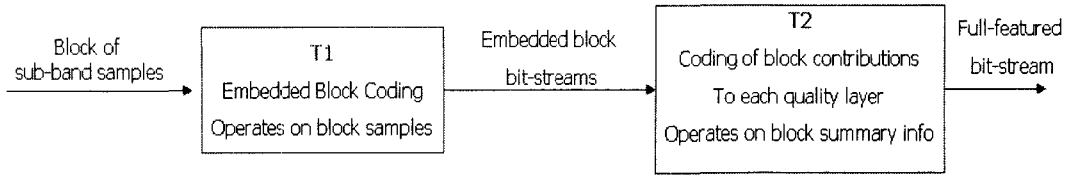


그림 2. EBCOT 코딩 과정

(LSB : Least significant Bit-Plane)순으로 인코딩된다. 3 가지의 코딩 패스는 Significant Propagation Pass, Refinement Pass 그리고 Clean Up Pass이다. 코딩 패스들은 WC 비트 모델링을 이용하여 각 계수의 컨텍스트를 선택한다. 패스 데이터는 컨텍스트 기반의 적응적 산술부호화 코더 (MQ-Coder)를 통해서 엔트로피 코딩된다.

Tier-2 코딩은 비율 제어를 수행한다. 기존 DCT 기반 알고리즘들처럼 JPEG2000 역시 먼저 양자화 과정에서 비율 제어를 대략적으로 수행한다. 하지만 포함될 코딩 패스를 선택하는 방법으로 좀 더 세밀한 비율 제어를 할 수 있다. 코드블록의 모든 패스는 코딩 패스의 끝인 절단점이 될 수 있다. 인코더는 이 절단점 뒤에 있는 모든 패스들을 무시한다. 최적화된 절단점을 결정하는 방법은 비율-왜곡(Rate-Distortion) 기준을 기반으로 한다. 목표 비트율에서 전체 왜곡을 최소화시키는 문제는 라그랑지 연산을 이용한 다음 방정식을 최소화하는 것과 같다.

$$\sum_i (R_i^{n_i} - \lambda D_i^{n_i}) \quad (1)$$

여기서  $R_i^{n_i}$ 과  $D_i^{n_i}$ 은 각각 절단점  $n_i$ 에서  $i$ 번째 코드블록의 코드 바이트 수와 평균 제곱 오차(MSE)로 표시되는 왜곡을 나타낸다. 절단점  $n_i$ 에서 R-D 기울기는 다음 방정식으로 계산된다.

$$S_i^{n_i} = \frac{\Delta D_i^{n_i}}{\Delta R_i^{n_i}} = \frac{\Delta D_i^{n_i-1} - \Delta D_i^{n_i}}{\Delta R_i^{n_i-1} - \Delta R_i^{n_i}} \quad (2)$$

$n_i$ 가 증가하면 절단점은 MSB에서 LSB로 내려간다. 기울기 값은  $\Delta D_i^{n_i}$ 가 점차 작아지므로 자연스럽게 단조 감소하는 성향을 갖는다. 라그랑지 곱셈 기술을 이용하면 최적화된 절단점을 선택할 수 있다. 이 절단점은 비율 제한에서 얻어진 상수  $\lambda$ 값과 크거나 같은 비율-왜곡 기울기를 갖는 모든 코딩 패스를 포함하게 된다[10].

## 2.2 ROI 코딩 방법

### 2.2.1 ROI의 필요성

ROI 코딩은 영상 전체를 전송 및 복원하기보다는 일부 특정 영역이나 사용자 위주의 ROI를 배경보다 우선처리를 한다. 이 기술은 영상의 다 해상도 해석을 이용한 점진적인 영상 코딩 기반 위에 달성된다. ROI 영상의 장점은 점진적 영상 전송 방법보다 압축률과 전송시간을 단축시킬 수 있을 뿐만 아니라 효율적인 메모리 관리가 가능하여 사용자의 다양한 요구에 응할 수 있다. [그림 3]은 0.125bpp에서 점진적인 영상과 ROI 영상으로 재구성된 "Lena" 영상을 나타낸다.

ROI 코딩 특징은 저 비트율에서 고화질 ROI를 볼 수 있고, 전체 영상 비트율과 ROI 비트율의 조정이 가능하며, 다양한 중요도 등급으로 다수 ROI를 지원한다. 또한 인코딩 시에 ROI를 정의하여 코딩하는 정적 ROI뿐만 아니라, 디코딩 시에 압축된 비트 스트림으로부터 ROI를 추출하는 코딩을 하는 동적 ROI도 지원한다.

### 2.2.2 Maxshift ROI 코딩 방법

이 방법은 양자화된 WC 중에 ROI WC와 배경 WC를 구분하여, 배경 WC 중에서 가장 큰 계수 값,  $s$ 를 구한 다음, ROI WC를  $s$ 보다 높은 비트-평면에 이동시키는 방법으로서 JPEG2000 Part1 표준이다. 식(3)은  $s$ 를 구하는 수식이고, 식(4)는 ROI 처리 후

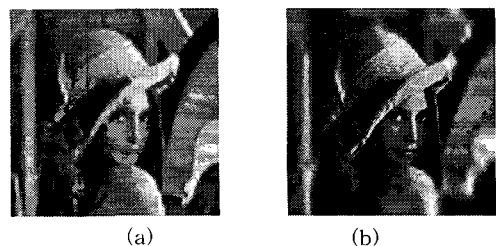


그림 3. (a) 점진적인 영상과 (b) ROI 영상

의 WC,  $a'(u,v)$ 을 구하는 수식이다.

$$s \geq \max(M_b) \tag{3}$$

$$a'(u,v) = \begin{cases} a(u,v), & M(u,v) = 0 \\ 2^s a(u,v), & M(u,v) = 1 \end{cases} \tag{4}$$

식(3)에서  $\max(M_b)$ 은 각 서브밴드에서 양자화된 배경 WC 중에 가장 큰 값을 의미한다. 식(4)에서  $M(u,v)$ 는 ROI 마스크 정보로서 WC가 ROI에 속하는 좌표인 경우는 1, 배경에 속하는 좌표인 경우는 0을 의미한다. 장점은 인코딩 시에 ROI 모양 정보 대신  $s$ 값만 가지게 되므로 코딩 효율이 좋다. 단점은 ROI가 모두 복원될 때까지 배경을 얻을 수 없다. 즉 ROI 중요도 조절이 불가능하다. 또한 다수 ROI 지원이 어렵다[11].

### 2.2.3 목시적 ROI 코딩 방법

이 방법은 EBCOT 기반의 코딩 방법으로서 디코딩 시에 ROI를 정의하여 코딩하는 동적 ROI 코딩 방법이고, JPEG2000 Part1 표준이다. EBCOT에서 각 품질 레이어는 코드블록들(혹은 패킷, 프리신트)의 임베디드 비트 스트림으로부터 임의의 공헌도를 포함한다. 따라서 우선처리는 각 코딩 패스에서 손실률을 조절한 후, PCRD(Post-Compression Rate-Distortion) 최적화 알고리즘을 다시 수행한다. 전체 손실 최소화에 의해서 코드블록 공헌도를 할당하기 때문에, 목시적 ROI는 손실 감소와 ROI가 일치하도록 코드블록 공헌도를 할당한다. 식(5)은 손실 계산방법을 나타낸다.

$$D_j^{n_i} = \begin{cases} W_{ROI} w_b \sum_{u,v \in B_j} (\hat{a}^n(u,v) - a(u,v))^2, & ROI \text{ 코드블록} \\ w_b \sum_{u,v \in B_j} (\hat{a}^n(u,v) - a(u,v))^2, & \text{그 외} \end{cases} \tag{5}$$

식(5)에서  $W_{ROI}$ 는 가중치이고,  $D_j^{n_i}$ 은  $n_i$ 에서의 가중 MSE 손실이고,  $w_b$ 는  $B_j$ 을 포함하는 서브밴드의 가중치이고,  $B_j$ 은  $j$ 번째 코드블록이고,  $a(u,v)$ 은 WC이고,  $\hat{a}^n(u,v)$ 은  $n_i$ 로 양자화된 WC이고,  $n_i$ 은 절단점을 의미한다.

이 방법의 장점은 복잡도가 낮아 구현이 쉽고, 비트-평균 이동을 하지 않는다. 단점은 코드블록단위로 ROI가 지정되기 때문에 다각형 모양만 지원하다. 또한 ROI 코드블록 안에 포함된 배경 WC도 ROI 처리를 함으로서 ROI 코딩 성능이 떨어진다.

### 2.2.3 수정된 목시적 ROI 코딩 방법

수정된 목시적 ROI는 알고리즘 복잡도 없이 ROI 코드블록에 포함된 배경 WC의 우선권을 줄여 목시적 방법을 보완한 방법이다. ROI 코드블록 내의 배경 WC의 우선권 조절은 배경 WC의  $k$  LSB만큼 절단하는 방법으로 실현된다. 식(6)은 ROI 처리 전의 ROI 코드블록내의 WC인  $a(u,v)$ 을 우선권 조절 후의 WC인  $\bar{a}(u,v)$ 로 변환하는 식이다.

$$\bar{a}(u,v) = \begin{cases} \text{sign}[a(u,v)] \left\lfloor \frac{|a(u,v)|}{2^k} \right\rfloor 2^k, & BG \text{ 계수} \\ a(u,v), & ROI \text{ 계수} \end{cases} \tag{6}$$

매개변수  $k$ 는 인코더에서 지정되며, 조절이 가능하다.  $k$  값이 클수록 ROI 코드블록내의 배경 WC의 우선권이 낮아진다. 장점은 WC 단위로 ROI 처리가 가능하며, 전체 ROI 전송이 목시적 ROI 방법보다 약 24% 속도가 빠르다는 것이다. 그리고 ROI 크기가 작을수록 무손실 재구성 비트율도 낮아진다. 단점은 ROI 코드블록에 포함된  $k$  LSB의 절단 때문에, 무손실 ROI 코딩 방법과는 호환되지 않는다[6].

### 2.3 새로운 목시적 ROI 코딩의 필요성

최상의 ROI 서비스는 다양한 비트율(특히 저 비트율)에서 ROI 중요도에 따라 고품질 ROI와 배경을 제공할 뿐만 아니라, 임의의 ROI 모양을 지원해야 한다. 목시적 방법은 코드블록 단위로 우선처리를 함으로서 임의의 ROI 모양을 지원하지 않고, ROI 코드블록 내의 배경 WC의 우선처리로 인해 ROI 품질 저하를 초래한다. 수정된 목시적 ROI 코딩 방법은 ROI 코드블록내의 배경 WC의  $k$  LSB를 절단하여 ROI 품질은 향상시켰지만, 무손실 코딩에 호환이 안되는 단점이 발생한다. 또한 ROI 코드블록 내의 배경 WC의 절단을 위해 모든 WC를 비교해야 하는 계산 복잡도가 발생한다. 이러한 문제들은 ROI 코드블록 내의 배경 WC의 수가 많을수록 커진다. 그리고 기존 방법은 인코더와 디코더가 서로 다른 시스템에 존재할 경우에 ROI 코딩을 위해 영상 도메인에서의 임의의 ROI 모양 정보를 전송하는데, 이 모양 정보는 1bpp로 구성된 영상 크기의 정보로서 비교적 많은 양이기 때문에 코딩 성능에 영향을 미친다.

결국 최상의 서비스를 위해 인코더로 전송되는 ROI 모양 정보를 줄이고 ROI 코드블록 내의 배경

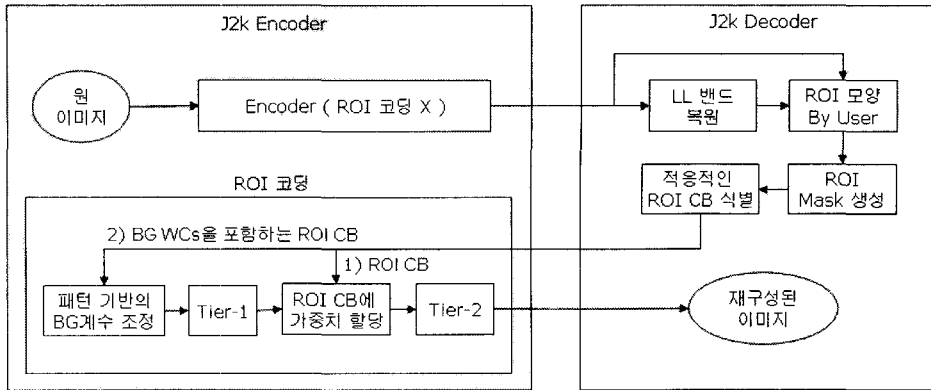


그림 4. 적응적인 목시적 ROI 코딩 방법의 구성도

WC의 우선처리와 삭제 문제의 개선과 비교 연산에 대한 계산 복잡도를 줄여 빠른 ROI 코딩을 할 수 있는 적응적 방법이 요구된다.

### 3. 적응적 ROI 코딩 방법

본 장에서는 기존 목시적 방법과 수정된 목시적 방법을 보완한 적응적 목시적 ROI 코딩 알고리즘을 소개한다.

#### 3.1 코딩 방법의 내부 구조

[그림 4]는 적응적인 목시적 ROI 코딩 방법의 구성도를 나타낸다. 인코더와 디코더는 하나의 시스템에 함께 존재할 수도 있고, 별도의 시스템에 각각 존재하여 네트워크로 연결될 수도 있다.

제안된 방법은 동적 ROI 방법이므로 원 영상은 ROI를 포함하지 않은 영상이 실험으로 사용된다. 먼저, 인코더로부터 수신된 영상의 기본적인 정보와 LL 밴드 내용을 복원하고, 이를 토대로 사용자는 영상 도메인에서 임의의 ROI 모양을 정의한다. 둘째, 각 분해 레벨의 서브밴드에 따라 ROI 마스크를 생성한다. 셋째, 패턴기반의 적응적인 ROI 코드블록 식별하여 식별 정보를 구하고, 이 정보를 인코더로 전송한다. 넷째, 인코더에서는 수신된 ROI 코드블록 식별 정보를 기반으로 적응적인 우선처리를 한다.

완전한 ROI 코드블록의 우선처리는 ROI 코드블록에 가중치를 할당한 후에 Tier-2과정을 거쳐 영상을 재구성하고, 배경 WC를 포함하는 ROI 코드블록의 우선처리는 패턴 기반의 배경 WC 우선권 조정,

Tier-1, ROI 코드블록에 가중치를 할당, Tier-2 과정을 거쳐 영상을 재구성한다.

#### 3.2 임의의 ROI 모양을 위한 ROI 마스크 생성

사용자에 의해 영상 도메인에서 임의의 ROI 모양이 정의되면, 1 비트로 구성된 영상 크기의 ROI 마스크 정보가 생성된다. 이 정보는 IDWT(Inverse DWT)에 의해 각 분해 레벨에서의 ROI 모양 정보를 구하여 최종 ROI 마스크를 구한다. 최종 ROI 마스크 생성 과정은 우선 영상 영역에서의 ROI 마스크와 마지막 IDWT에 의해 2개의 서브밴드 안에 어떤 위치의 WC들이 ROI WC로 재구성하기 위해 사용되었는지를 추적하게 된다. 그런 후에, 두 서브밴드에서의 ROI 마스크와 마지막 이전의 IDWT에 의해 각각 2개의 서브밴드 안에 어떤 위치의 WC들이 ROI WC로 재구성하기 위해 사용되었는지를 추적하게 된다. 이렇게 하여 각 분해레벨에서 모든 서브밴드 내에 어떤 위치의 WC들이 ROI WC로 재구성하기 위해 사용되었는지를 추적한다[그림 5].

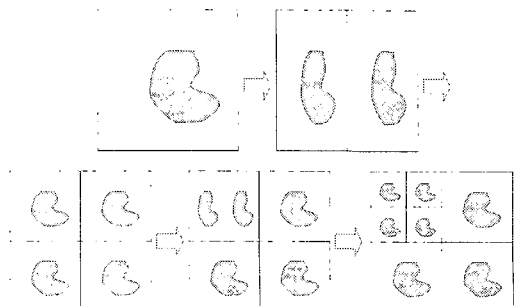


그림 5. ROI 마스크 생성

만약 하나의 분해레벨에서 하나의 행이나 열의 WC가 1D 신호라고 한다면, 원 샘플들을  $X(2n)$ 와  $X(2n+1)$ 로 표시하고, L 서브밴드와 H 서브밴드에 속하는 WC를 각각  $L(n)$ 과  $H(n)$ 로 표시한다. 그러면 ROI 마스크는  $X(2n)$ 와  $X(2n+1)$ 의 계산에 필요한  $L(n)$ 과  $H(n)$ 를 추적함으로써 구할 수 있다. 예를 들어 5/3필터에 대한 ROI 마스크를 추적하기 위한 IDWT는 식(7)에 나타낸다. 식(7)에 의하여,  $X(2n)$ 를 재구성하기 위해 필요한 WC는  $L(n)$ ,  $H(n-1)$  그리고  $H(n)$ 이다: 따라서  $X(2n)$ 가 ROI 마스크 안에 있다면  $L(n)$ ,  $H(n-1)$  그리고  $H(n)$ 가 필요하다. 동일하게  $X(2n+1)$ 가 ROI 마스크 안에 있다면  $L(n)$ ,  $L(n+1)$ ,  $H(n-1)$ ,  $H(n)$  그리고  $H(n+1)$ 가 필요하다.

$$X(2n) = L(n) - \frac{H(n-1) + H(n)}{4} \quad (7)$$

$$X(2n+1) = \frac{L(n) + L(n+1)}{2} + \frac{-H(n-1) + 6H(n) - H(n+1)}{8}$$

이와 같은 방법으로 생성된 ROI 마스크 정보는 [그림 6]과 같이 각 분해레벨에서 서브밴드 안에 ROI 코드블록, 배경 코드블록 그리고 일부 배경 WC를 포함하는 ROI 코드블록으로 구성된다.

### 3.3 적응적인 ROI 코드블록 식별

기존 방법의 단점인 ROI 코드블록 내의 배경 WC의 우선처리와 손실 문제를 보완하기 위하여 ROI 코드블록 내의 배경 WC의 분포된 모양에 따라 6 가지 패턴 중에 하나로 코드블록을 식별한다. [표 1]은 제안된 방법에서 사용하는 6 가지 패턴을 나타낸다.  $P_0$ 은 배경 코드블록을 의미하고,  $P_1$ 은 순수 ROI 코드블록을 의미하고,  $P_2, P_3, P_4, P_5$ 는 배경 WC를 포함하는 ROI 코드블록을 의미한다.

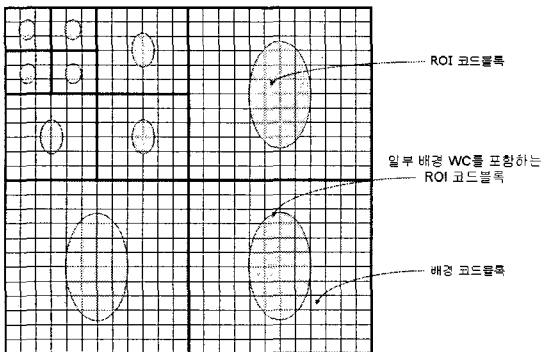


그림 6. ROI 마스크

표 1. 6 가지 패턴 모양

패턴번호(P)	0	1	2	3	4	5
모양						
의미	BG	ROI	BG WCs를 포함하는 ROI CB			

패턴 기반의 코드블록 식별 과정은 ROI 마스크 정보에서 코드블록 내의 배경 WC의 포함된 양과 임계값에 의해  $P_0$ 혹은  $P_1$ 인지를 먼저 식별한다. 즉 코드블록 내의 배경 WC의 포함된 양이 최대 임계값보다 크면  $P_0$ 로 식별하고, 최소 임계값보다 적으면  $P_1$ 로 식별한다.

본 논문에서는 최대 임계값을 90%, 최소 임계값은 10%로 사용하였다. 각 임계값은 100%와 0%에 차이가 많을수록 ROI와 배경 품질에 영향을 미친다. 이 값들은 조절이 가능하다.  $P_0$ 혹은  $P_1$ 로 식별되지 않으면, 코드블록을 [그림 7]의 (a)와 같이 A, B, C 세 방향으로 스캔하여 각 방향에서 ROI 혹은 배경 WC로 시작하여 배경 혹은 ROI WC를 찾고, 이 위치들을 a, b, c라고 한다. 만약 A, C 방향의 스캔에서 ROI 혹은 배경 WC가 없으면, A', C' 방향으로 다시 스캔하여 a, c를 찾는다. 스캔 중에 배경 WC로 시작하여 ROI WC를 찾으면, 해당 코드블록은  $P_2$ 와  $P_3$  중에 하나의 패턴이 되고, ROI WC로 시작하여 배경 WC를 찾으면,  $P_4$ 와  $P_5$  중에 하나의 패턴이 된다.

그런 후에 a, b, c 좌표를 이용하여 기울기를 구한

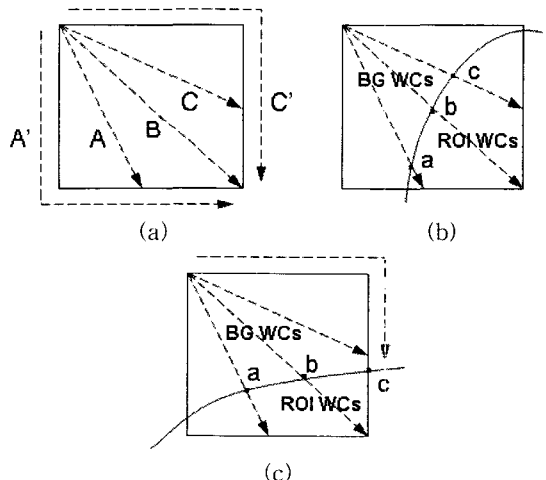


그림 7. 코드블록 식별 과정: (a) 스캔 순서, (b) 3번 패턴, (c) 2번 패턴

다. 이 기울기의 절대 값이 1보다 작으면  $P_2$  혹은  $P_4$ 이고, 크면  $P_3$  혹은  $P_5$ 이다. 마지막으로 각 패턴에서 폭을 구하게 되는데,  $P_2$ 의 폭은 코드블록 전체 폭과  $\text{MinY}(a, b, c)$ 의 차이이고,  $P_3$ 의 폭은 코드블록 전체 폭과  $\text{MinX}(a, b, c)$ 의 차이이고,  $P_4$ 의 폭은  $\text{MaxY}(a, b, c)$ 이고,  $P_5$ 의 폭은  $\text{MaxX}(a, b, c)$ 이다.

[그림 8]은 목시적 방법, 수정된 목시적 방법 그리고 적응적인 목시적 ROI 방법에서 식별된 코드블록을 나타낸다. 점선은 사용자의 의해 지정된 ROI 모양을 의미하고, 흰색 부분은 배경, 회색은 ROI 처리되는 부분을 의미한다.

코드블록 식별 정보는 패턴(3 비트)과 폭(5 비트)

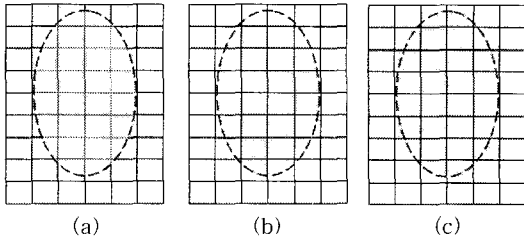


그림 8. ROI 코드블록 식별: (a) 목시적 ROI, (b) 수정된 목시적 ROI, (c) 적응적인 목시적 ROI

정보로 구성된다. 패턴 정보는 [그림 7]의 각 패턴의 번호를 의미한다. 폭 정보는 ROI 코드블록 내에서 ROI WC가 차지하는 양을 의미하며, 32X32 코드블록인 경우는 5 비트로 표현을 하고, 64X64 코드블록인 경우는 폭 정보를 1/2로 줄여서 표현한다. 이렇게 코드블록 당 한 바이트로 구성된 코드블록 식별 정보는 ROI 코딩을 위해 인코더로 전송을 하게 된다.

### 3.4 ROI 처리의 2 가지 모드

인코더에서는 디코더로부터 수신된 코드블록 식별 정보를 기반으로 적응적인 ROI 처리를 한다.  $P_1$ 인 코드블록은 목시적 방법으로 우선처리를 하고  $P_2, P_3, P_4, P_5$ 인 코드블록은 패턴 기반의 수정된 목시적 방법으로 우선처리를 한다. 패턴 기반의 수정된 목시적 방법은 기존 수정된 목시적 방법과 우선처리 방법은 동일하지만, ROI 코드블록을 식별하는 방법에서 많은 차이가 있다. 즉 수정된 목시적 방법은 ROI 코드블록 내의 배경 WC를 식별하기 위해 인코더에서 ROI 코드블록 내의 모든 WC를 비교한다. 패턴 기반의 수정된 목시적 방법은 디코더에서 ROI 코드블록 내의 일부 WC와 비교를 통해 패턴 기반으로

```

01: static mode {
02:   generate ROI mask in the wavelet domain
03:   identify ROI code blocks }
04: dynamic mode {
05:   code LL frequency band and send the resulting bit-stream to decoder
06:   accept co-ordinates of user-defined ROI
07:   generate ROI mask in the wavelet domain
08:   identify ROI code blocks
}
09: for each CB (n) { // CB : code block
    Pnum = GetPatternNum(GetCBInfo(n))
10:   if 1 < Pnum AND CB ∉ lowest-lowest subband
11:     ReducePriority(Pnum, GetCBInfo(n))
12:   for bitplane = MSB : LSB {
13:     multipass coding {
14:       encode each pass
15:       calculate coding pass rate R
16:       calculate coding pass distortion estimate D
17:       if code block ∈ ROI code block then D = D * WROI
    } //end
  } //end bit-plane
} //end code block
19: rate control to pack bit stream into quality-layered structure
    
```

그림 9. 적응적인 목시적 관심영역 코딩 알고리즘

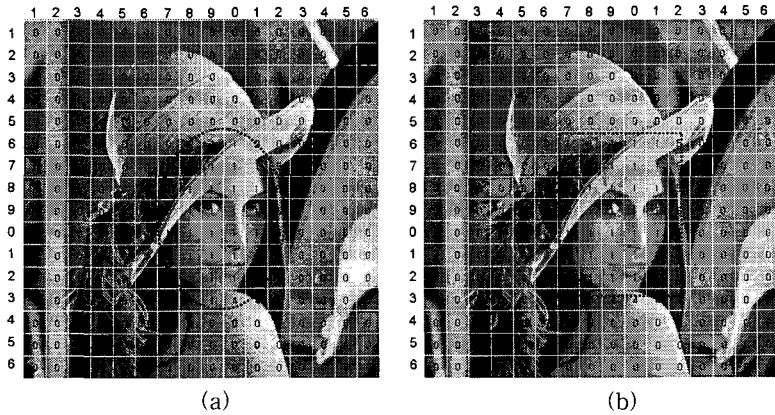


그림 10. ROI 모양에 따른 ROI 코드블록 식별: (a) 타원, (b) 직사각형

코드블록 식별하고, 이를 기반으로 인코더에서 우선 처리를 한다. [그림 9]의 알고리즘은 제안된 적응적인 목시적 ROI 코딩 방법을 나타내고 있다.

#### 4. 실험 및 평가

##### 4.1 실험 방법

실험은 ROI 코딩을 하지 않은 EBCOT와 Max-shift, 목시적 방법, 수정된 목시적 방법 그리고 제안한 방법을 사용한다. 목시적 방법, 수정된 목시적 방법, 제안된 방법에서는  $W_{ROI}$ 의 값을 4096로 설정하고, 동일한 실험 조건을 위해 모두 사각형 ROI를 이용

하고, 위치는 영상의 중앙 부분으로 하고, 크기는 전체 영상의 1/4를 차지한다. 실험 영상은 [표 2]와 같으며, 모두 흑백 영상(8bpp)이다. 실험 결과는 6개의 실험 영상을 사용하여 나온 결과를 평균한 것이다.

##### 4.2 실험 예

[그림 10]는 ROI 모양에 따른 ROI 코드블록 식별을 나타낸다. 각 네모는 32\*32 크기의 코드 블록이고, 번호는 식별된 코드블록의 패턴 번호 이고, 점선의 타원과 사각형은 ROI 모양이다. 코드블록 식별은 3.3절의 적응적인 ROI 코드블록 식별 방법에 의해서 패턴 번호와 폭 정보가 구해지고, 식별 정보는 하나의 코드블록에 1 바이트로 처리된다.

##### 4.3 비교평가

[표 3]은 ROI가 있는 lena 영상을 0.0625bpp부터 1.0bpp까지의 비트율로 복원한 ROI의 PSNR값을 나타내고, Maxshift와 제안된 방법이 기존의 ROI 방법들보다 우수한 성능을 보인다.

[표 4]는 ROI의 무손실 재구성을 위해 요구되는 비트율을 나타낸다. ROI의 크기가 전체 영상의 1/4

표 2. 실험 영상

파일명	해상도	종류
lena	512 × 512	자연영상
mountain	640 × 480	자연영상
library	464 × 352	복합영상
boat	512 × 512	자연영상
satellite	1417 × 793	항공영상
ultrasound	998 × 1614	초음파영상

표 3. 다양한 방법에 대한 ROI PSNR의 비교

비트율	EBCOT	목시적	Maxshift	수정된 목시적	제안한 방법
0.0625	23.413	24.200	28.085	24.516	24.968
0.125	25.056	27.299	30.689	28.208	30.183
0.25	27.897	30.919	33.976	32.603	33.217
0.5	31.620	35.897	39.014	38.117	38.688
1.0	37.871	43.38	47.733	37.795	38.127



표 4. ROI 무손실 재구성을 위한 비트율 비교

ROI 코딩 방법	비트율	
	25% ROI	8% ROI
EBCOT	2.960	2.960
목시적	1.909	1.113
Maxshift	1.361	0.554
수정된 목시적	1.449	0.671
제안한 방법	1.407	0.642

인 경우 ROI의 완전한 재구성은 EBCOT는 2.960bpp, 목시적 방법은 1.909bpp, 수정된 목시적 방법은 1.449bpp에서 이루어지는 반면에, 제안된 방법은 1.407bpp에 이루어진다. 그리고 [표 4]처럼 ROI의 크기가 작을수록 ROI의 무손실 재구성 비트율도 낮아진다. [그림 11]은 제안된 방법에 의해 0.125, 0.5, 1.0, 2.0bpp로 재구성된 lena 영상을 나타낸다.

[그림 12]은 0.25bpp에서 다양한 코딩 방법을 사용하여 재구성된 영상들을 보인다. [그림 10]과 같이 다른 EBCOT 기반의 ROI 방법보다 제안된 방법의 영상 품질이 우수하게 나타난다.

[그림 13]는 저 비트율에서 ROI와 전체 영상의 PSNR 값을 통해 다양한 ROI 코딩 방법의 성능 비교를 나타낸다. 제안된 방법은 저 비트율에서 전체 영상에 대한 압축 성능이 뛰어나다. 실험 영상 외의 다른

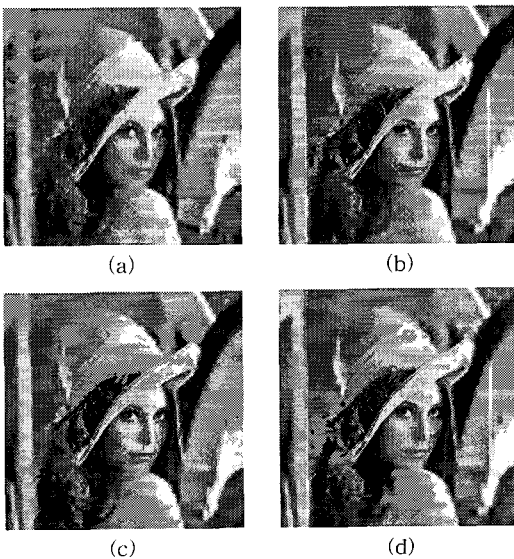


그림 11. 제안된 ROI 코딩 방법을 이용한 재구성: (a) 0.125bpp, (b) 0.5bpp, (c) 1.0bpp, (d) 2.0bpp

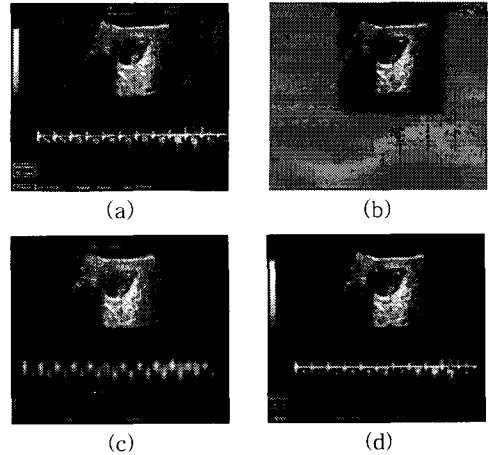


그림 12. 0.25bpp에서 다양한 방법으로 재구성된 영상: (a) Original, (b) Maxshift, (c) Modified, (d) Adaptive

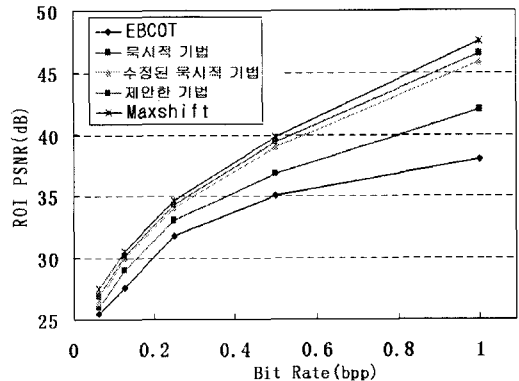


그림 13. 다양한 ROI 코딩 방법을 사용한 전체 영상의 성능 비교

영상에서도 비슷한 결과를 관찰할 수 있다.

제안된 방법도 수정된 목시적 방법과 마찬가지로 ROI 코드블록 내의 배경 WC를 삭제하기 때문에 JPEG2000 표준 ROI의 무손실 코딩과는 호환이 안 된다.

### 5. 결론 및 향후 과제

본 논문에서는 기존 EBCOT 기반의 ROI 방법들을 보완한 적응적인 목시적 ROI 코딩 방법을 제안하였다. 기존의 방법은 ROI와 배경이 섞여 있는 경계 영역에 있는 픽셀들을 처리할 때, ROI가 아닌 부분이 ROI로 인식되는 문제와 배경 부분을 손실 처리하는 문제가 있었고, 또한 마스크 정보의 양도 많았다. 이

문제를 개선하기 위하여, 제안된 방법은 패턴 기반의 코드블록 식별을 수행하고, ROI 마스크 생성 및 ROI 코드블록 식별 과정을 디코더에서 수행하여 마스크 정보의 양을 줄였다. 기존 EBCOT 기반의 ROI 방법들과 비교 실험을 통해 제안된 방법이 3%정도 압축 성능 면에서 뛰어난 것을 확인하였다. ROI 뿐만 아니라 전체 영상에서의 압축 성능도 3%정도 뛰어난 것을 볼 수 있었다. 향후 과제는 이 기법을 정적 ROI 기법에도 적용하여 실험할 것이고, 여러 ROI도 지원할 수 있게 수정하여 실험할 것이다.

### 참 고 문 헌

[1] C. Christopoulos, "JPEG2000 Verification Model 8.5(Technical description)," *ISO/IEC JTC 1/SC 29/WG 1 WG 1 N1878*, Sep. 2000.

[2] ISO/IEC International Standard 15444-1, ITU Recommendation T.800, "JPEG2000 Image Coding System," 2000.

[3] C.Christopoulos, A.Skodras, and T.Ebrahimi, "The JPEG2000 still image coding system: An overview," *IEEE Transactions on Consumer Electronics*, Vol. 46, pp. 1103-1127, Nov. 2000.

[4] H. S. Kong, A. Vetro, T. Hata, and N. Kuwahara, "Fast Region-of-Interest Transcoding for JPEG 2000 Images," *Mitsubishi Electric Research Laboratories*, 2005.

[5] D. S. Taubman and M. W. Marcellin, "JPEG 2000 Image Compression Fundamentals," *Standard and Practice*, pp. 637-641, 2002.

[6] H. Yang, M. Long and H. M. Tai, "Region-of-Interest Image Coding Based on EBCOT," *IEE Proceedings-Vision, Image, and Signal Processing*, Vol. 152, Issue 5, pp.590-596, Oct. 2005.

[7] R. Rosenbaum and H. Schumann, "Flexible, Dynamic and Compliant Region Of Interest Coding in JPEG2000," *IEEE ICIP*, Sep. 2002.

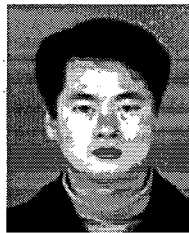
[8] V. Sanchez, A. Basu, and M. K. Mandal, "Prioritized Region Of Interest Coding in JPEG2000," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 14,

No. 9, Sep. 2004.

[9] D. S. Taubman, KaKadu S/W V2.2, <http://www.kakadusoftware.com/>, 2002.

[10] M. Rabbani and R. Ioshi, "An Overview of the JPEG2000 Still Image Compression Standard," *Signal Processing : Image Communication*, Vol. 17, pp. 3-48, 2002.

[11] R. Grosbois, D. Santa-Crus, and T. Ebrahimi, "New Approach to JPEG2000 Compliant ROI Coding," *Proc. of the SPIE*, Aug. 2001.



격교육

### 강 기 준

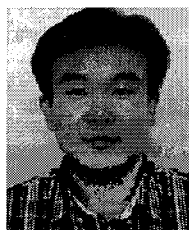
2000년 경상대학교 컴퓨터과학  
과 학사  
2002년 경상대학교 컴퓨터교육  
과 석사  
2003년~현재 경상대학교 컴퓨터  
과학과 박사과정  
관심분야 : MPEG, JPEG2000, 원



과학과 교수

### 이 부 권

1972년 경상대학교 농경제학과  
학사  
1978년 미시건주립대학 대학원  
시스템 공학전공 석사  
2001년 경남대학교 대학원 컴퓨  
터공학과 박사  
1980년~현재 경상대학교 컴퓨터  
2001년~현재 경상대학교 컴퓨터정보통신연구소원  
관심분야: 시뮬레이션, 멀티미디어, JPEG2000



### 서 영 권

1987년 경상대학교 전산통계학  
과 학사  
1989년 숭실대학교 대학원 전자  
계산학과 석사  
1997년 숭실대학교 대학원 전자  
계산학과 박사  
1989년~1992년 삼보컴퓨터 대리  
1997년~현재 경상대학교 컴퓨터교육과 부교수  
2001년~현재 경상대학교 컴퓨터정보통신연구소원  
관심분야 : 멀티미디어통신, 영상인식, JPEG2000