

# JPEG2000 이미지에서 적응적 코드블록 판별 알고리즘을 이용한 동적 고속 관심영역 코딩 방법

강 기 준<sup>†</sup> · 서 영 건<sup>\*\*</sup> · 박 재 흥<sup>\*\*\*</sup> · 유 창 열<sup>\*\*\*\*</sup> · 박 순 화<sup>\*\*\*\*\*</sup> · 이 점 숙<sup>\*\*\*\*\*</sup> · 이 부 권<sup>\*\*\*\*\*</sup>

## 요 약

본 논문에서는 명시적 ROI 코딩 방법과 수정된 명시적 ROI 코딩 방법을 보완한 JPEG2000 이미지에서 적응적 코드블록 판별 알고리즘을 이용한 동적 고속 관심영역 코딩 방법을 제안한다. 제안한 방법에서는 코드 블록 판별 시간을 줄이기 위하여 ROI 모양의 특징을 고려하여 몇 가지의 경계 모양을 만들고, 이 모양 패턴으로 한정하여 코드블록 판별 알고리즘을 만들었다. 그리고 ROI 임계값과 배경 임계값을 이용하여 ROI 코드블록 내의 배경 웨이블릿 계수의 포함율에 따라 적응적인 코드블록 판별을 함으로서 ROI 코드블록 내의 배경 웨이블릿 계수의 우선적 처리와 손실 문제를 개선하였다. 또한 ROI 코드블록 내의 배경 웨이블릿 계수의 우선권 조절은 웨이블릿 계수 단위로 처리하는 기존의 방법과는 달리 패턴을 기반으로 하여 일괄처리를 함으로서 빠른 ROI 코딩이 가능하다. 제안한 방법의 유용성을 보이기 위해 기존의 방법들과 비교 실험한 결과, 제안한 방법이 기존의 방법들과 성능 평가에 있어서 큰 차이가 없으면서 처리 속도는 매우 빠름을 확인하였다.

키워드 : 관심영역, 정지영상, 압축표준

## A Fast and Dynamic Region-of-Interest Coding Method using the Adaptive Code-Block Discrimination Algorithm in JPEG2000 Images

Ki-Jun Kang<sup>†</sup> · Yeong-Geon Seo<sup>\*\*</sup> · Jae-Heung Park<sup>\*\*\*</sup> · Chang-Yeul Yoo<sup>\*\*\*\*</sup> ·  
Soon-Hwa Park<sup>\*\*\*\*\*</sup> · Jum-Suk Lee<sup>\*\*\*\*\*</sup> · Bu-Kwon Lee<sup>\*\*\*\*\*</sup>

## ABSTRACT

In this paper, we propose a fast and dynamic Region-of-Interest coding method using the adaptive code-block discrimination algorithm in JPEG2000 images which complements the implicit ROI coding method and the modified implicit ROI coding method. For reducing the time of discriminating the code block, the proposed method estimates the characteristics of the shape of ROI and makes the shape of boundaries, and classifies the patterns of each code block. The method improves the preferred processing and loss of wavelet coefficients of background within the ROI code blocks by adaptively classifying the code blocks with the percentage of content of the wavelet coefficients using the thresholds of ROI and background. Also, the priority control of wavelet coefficients of background within ROI code block supports the rapid ROI coding by processing in batch based on patterns unlike the existing methods that process with unit of wavelet coefficients. To show the usefulness of this method, we compared this to the existing methods. There is no difference in performance, but we confirmed very speedy in processing time.

Key Words : JPEG2000, ROI, EBCOT

## 1. 서 론

인터넷과 멀티미디어 기술의 발전으로 이미지 정보는 의

료진단, 웹 브라우징, 이미지 데이터베이스 그리고 컴퓨터 통신 등의 광범위한 응용분야에서 활용되고 있다[1,2]. 이러한 이미지 정보는 저장 및 전송하기 위해서 많은 메모리와 넓은 대역폭을 요구하기 때문에 신속한 이미지 처리가 어려운 실정이다. 이를 해결하기 위하여 전체 이미지를 전송 및 복원하기보다는 이미지의 일부 특정 영역이나 사용자 위주의 관심영역에 대한 우선적 처리가 요구되고 있다. 이러한 우선적 처리는 JPEG2000에서 표준 ROI(region of interest) 코딩 방법들을 비롯하여, 이를 보완한 새로운 ROI 코딩 방

† 정 회 원 : 경상대학교 컴퓨터과학과 박사  
 \*\* 종신회원 : 경상대학교 컴퓨터교육과 부교수  
 \*\*\* 정 회 원 : 경상대학교 컴퓨터과학과 정교수  
 \*\*\*\* 정 회 원 : 경남도립 남해전문대학 컴퓨터정보과 부교수  
 \*\*\*\*\* 준 회 원 : 경상대학교 컴퓨터과학과 박사수료  
 \*\*\*\*\* 정 회 원 : 경상대학교 컴퓨터과학과 정교수(교신저자)  
 논문접수 : 2007년 2월 20일, 심사완료 : 2007년 8월 16일

법들에 대한 연구가 최근까지 활발하게 이루어지고 있다 [3,4]. ROI 코딩 방법은 언제 ROI를 지정하느냐에 따라서 정적 ROI와 동적 ROI 코딩 방법으로 나뉜다. 정적 ROI 코딩 방법은 압축 과정에서 ROI 모양을 지정하여 ROI 코딩을 하는 방법으로 대부분의 응용에서 적용이 가능하다. 동적 ROI 방법은 복원 과정에서 사용자 위주의 ROI 모양을 지정하여 ROI 코딩을 하는 방법으로서 ROI 코딩이 되어 있지 않은 압축된 비트스트림으로부터 임의의 ROI를 추출하여 우선적 처리를 하는 방법이다[6]. 이 방법에서는 표준 방법인 목시적 방법과 이를 보완한 수정된 목시적 방법이 있다. 하지만 기존의 동적 ROI 방법은 ROI 코드블록(code-block) 내의 배경 웨이블릿 계수(wavelet coefficient)에 대한 우선적 처리 문제와 손실 문제 그리고 ROI 코드블록을 판별하기 위하여 많은 스캔과정을 필요로 하여 코딩 속도를 저하시키는 문제가 발생한다. 본 논문에서는 이러한 문제점들을 보완한 JPEG2000 이미지에서 적응적 코드블록 판별 알고리즘을 이용한 동적 고속 관심영역 코딩 방법을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서는 관련연구로 JPEG2000 압축 알고리즘과 ROI 코딩 방법에 대해 알아본다. 3장에서는 본 논문에서 제안하는 패턴 기반의 관심영역 코딩 방법을 설명한다. 4장에서는 실험을 통하여 제안방법의 유효성을 검증한다. 마지막으로 5장에서는 결론 및 향후 연구를 제시한다.

2. 관련연구

2.1 JPEG2000 이미지 압축 표준

JPEG2000은 ISO/IEC 산하의 JTC1/S29/WG1 그룹에서 새롭게 개발된 이미지 압축 표준이다. 이 표준은 기존의 표준기술보다 올 왜곡, 이미지성능 면에서 우수성을 제공할 뿐 아니라 현재의 표준들이 언급할 수 없는 특징이나 기능들을 제공한다.

JPEG2000의 대표적인 특성은 무손실/손실 압축, 무손실 코딩에서의 내포된 손실, 화소 정확성이나 해상도에 의한 진보적인 전달, 비트 오류 그리고 ROI 코딩 등이 있다. JPEG2000의 필요성이 강조되는 다양한 응용으로는 인터넷, 컬러 팩스, 인쇄, 스캐닝, 디지털 사진, 이동식 적용, 의학

이미지, 전자 도서관과 전자상거래 등이 있다[1,2]. (그림 1)의 (a)처럼 JPEG2000 인코딩 과정은 원 이미지 데이터를 가지고 전 처리 과정을 거쳐 이산 웨이블릿 변환(discrete wavelet transform)을 하고, 변환된 웨이블릿 계수는 양자화한 후 엔트로피 과정을 통해 코드 열을 생성한다. 디코딩 과정은 인코딩 과정을 역으로 한다.

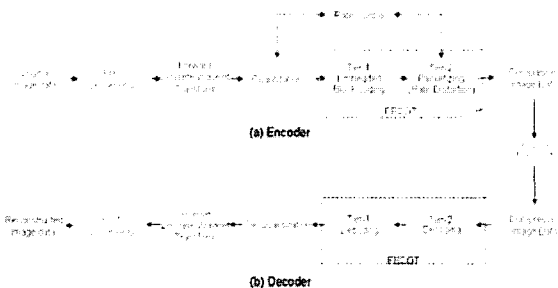
2.2 ROI 코딩 방법

ROI 코딩 방법은 JPEG2000의 주요 특징 중에 하나로서 이미지 내의 ROI를 배경보다 더 나은 품질로 빠르게 처리하는 기술로서 이미지 내의 특정 영역이 다른 영역보다 높은 중요성을 가지는 응용에 이용할 수 있다. ROI 이미지의 장점은 점진적 이미지 전송 방법보다 압축률과 전송시간을 단축시킬 수 있을 뿐만 아니라 효율적인 메모리 관리 등으로 인해 사용자의 다양한 요구에 응할 수 있다[3,7].

(그림 2)는 0.125bpp에서 점진적인 이미지와 ROI 이미지로 재구성된 "Lena" 이미지를 나타낸다. 이러한 ROI 코딩 방법은 정적 ROI 코딩 방법과 동적 ROI 코딩 방법으로 나뉜다.

정적 ROI 코딩 방법은 인코딩 과정에서 양자화된 웨이블릿 계수들 중에 사용자가 지정한 ROI 모양에 의해 ROI 웨이블릿 계수들과 배경 웨이블릿 계수들을 구분하여, ROI 중요도에 따라 웨이블릿 계수들의 비트-평면(bit-plane)을 이동시키는 방법이다. 이 방법에는 Maxshift[3], Scaling based[3], Maxshift-like[8], PSBS hift[9] 등이 있다. 대부분의 응용 분야에서는 정적 ROI 코딩 방법으로도 충분히 ROI 코딩 방법의 적용이 가능하지만, 인코딩 과정에서 ROI를 미리 알지 못하는 경우에는 디코딩 과정에서 ROI를 지정하게 된다. 이럴 경우 최초의 디코딩 과정에서 사용자가 원하는 ROI 모양 정보를 인코더 측에 넘겨주면 인코더 측에서는 그 정보를 토대로 ROI 코딩이 되어 있지 않은 압축된 비트스트림으로부터 임의의 ROI를 추출하여 우선 처리를 하게 한다. 이렇게 ROI 코딩을 실현하는 방법을 동적 ROI 코딩 방법이라고 한다. 대표적인 동적 ROI 코딩 방법에는 Implicit[3], Modified Implicit[5], Flexible and dynamic[10], Fast ROI Transcoding[6], Prioritized[11] 등이 있다.

Implicit[4] 방법은 Part1 표준으로서 코드블록 내에 ROI 웨이블릿 계수가 하나 이상 포함하면 ROI 코드블록으로 판



(그림 1) JPEG2000 블록 다이어그램



(a)

(b)

(그림 2) (a) 점진적인 이미지와 (b) ROI 이미지

별되어 우선적 처리한다. 이로 인해 ROI 코드블록에 포함된 배경 웨이블릿 계수도 우선적 처리를 하는 문제가 발생한다. 이를 보완하기 위해 Modified Implicit[4] 방법은 ROI 코드블록에 포함된 배경 웨이블릿 계수들의 우선권을 줄이기 위해 ROI 코드블록에 포함된 배경 웨이블릿 계수들의 비트-평면중에서  $k$  만큼의 LSB(Least Significant Bit-planes)를 삭제한 후에 ROI 코드블록으로 우선적 처리한다.

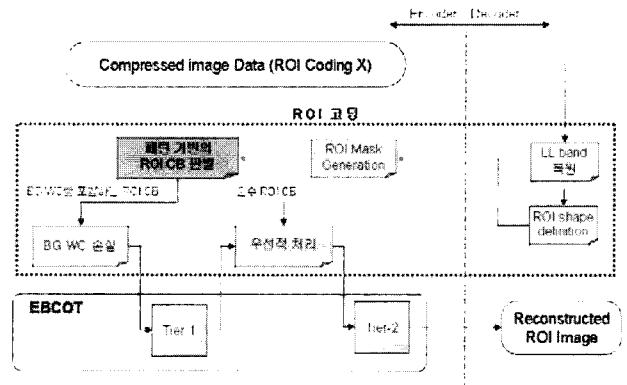
2.3 기존 동적 ROI 방법의 문제점

동적 ROI 코딩 방법은 저비트율에서 고 품질의 관심영역을 빠르게 서비스하기 위하여 정확하고 빠른 관심영역 코딩이 요구된다. 목시적 관심영역은 코드블록 단위로 우선처리를 함으로서 정확한 관심영역 모양 처리가 어렵고, 관심영역 코드블록 내의 배경 웨이블릿 계수의 우선적 처리로 인해 관심영역 품질 저하를 초래한다. 수정된 목시적 관심영역은 관심영역 코드블록 내의 배경 웨이블릿 계수에 대한 우선권 조절을 통해 관심영역 품질은 개선시켰지만 우선권 조절에 대한 계산복잡도로 인해 코딩 속도 저하시키는 문제가 발생한다. 또한 배경 웨이블릿 계수에 대한 우선권 조절을 하여도 여전히 배경 웨이블릿 계수의 우선적 처리 문제는 남아있다. 이러한 문제들은 관심영역 코드블록 내의 배경 웨이블릿 계수의 수가 많을수록 커진다. 기존 방법은 많은 스캔 과정을 통해 관심영역 코드블록을 판별하기 때문에 빠른 관심영역 코딩이 이루어지지 못한다.

3. 패턴 기반의 목시적 관심영역 코딩방법

(그림 3)은 제안한 방법의 전체 구성도이다. 인코더와 디코더는 하나의 시스템에 함께 존재할 수도 있고, 네트워크로 연결된 별도의 시스템에 각각 존재할 수도 있다. 인코더와 디코더가 별도의 시스템에 존재하는 경우에는 관심영역 마스크 생성과 관심영역 코드블록의 판별 과정을 디코더에서 처리하여, 적은 양의 관심영역 판별 정보를 인코더로 보낼 수 있어, 더욱 빠른 처리결과를 얻을 수 있다.

전체 구성을 살펴보면, (1) 관심영역을 결정하기 위해 관심영역 코딩이 되어 있지 않은 압축된 비트스트림으로부터 기본적인 정보와 LL 밴드 정보를 추출하여 디코더로 전송한다. (2) 디코더는 인코더로부터 수신된 대략적인 이미지 정보를 복원하고, 이를 토대로 사용자는 관심영역 모양을 지정하여 이미지 도메인에서의 관심영역 마스크를 생성한다. (3) 이미지 도메인에서의 관심영역 마스크와 IDWT(inverse DWT)를 사용하여 최종 웨이블릿 도메인에서의 관심영역 마스크를 구한다. (4) 서버밴드 내의 코드블록 단위의 마스크 정보에서 관심영역 웨이블릿 계수의 분포된 양에 따라 패턴기반의 관심영역 코드블록의 판별 과정을 거쳐 판별 정보를 구하고 이 정보는 관심영역 코딩을 위해 인코더로 전송한다. (5) 인코더에서는 수신된 관심영역 코드블록의 판별정보를 기반으로 우선적 처리를 한다. 즉 순수 관심영역 코드블록인 경우에는 관심영역 코드블록에 가장치를 할당한다



(그림 3) 제안한 방법의 전체 구성도

<표 1> 6 가지 패턴 모양

패턴번호(P)	P <sub>0</sub>	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>
모 양						
의 미	BG	ROI	BG WC를 포함하는 ROI			

후에 Tier-2과정을 거쳐 우선적 처리를 하고 배경 웨이블릿 계수를 포함하는 관심영역 코드블록인 경우에는 패턴 기반의 배경 웨이블릿 계수 우선권 조절, Tier-1, 관심영역 코드블록에 가장치를 할당, Tier-2 과정을 순서대로 거쳐 우선적 처리를 한다.

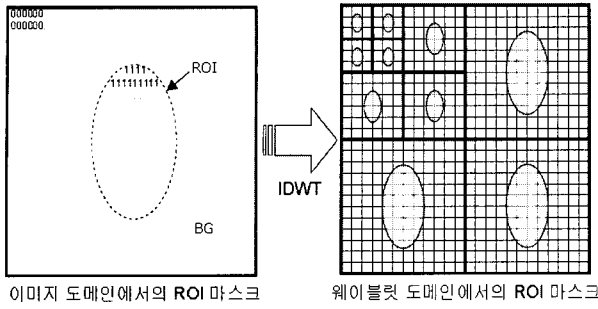
3.1 코드블록 단위의 ROI 패턴 분류

웨이블릿 도메인에서의 ROI 마스크는 (그림 4)와 같이 각 분해 레벨에서 서버밴드 안에 ROI 코드블록, 배경 코드블록 그리고 일부 배경 웨이블릿 계수를 포함하는 ROI 코드블록으로 구성된다.

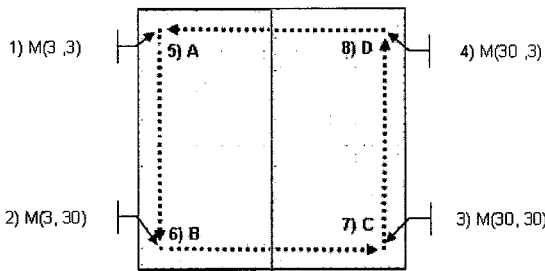
본 연구에서는 ROI 모양의 특징을 고려하여 몇 번의 스캔 과정과 간단한 연산을 통해서 코드블록당 1 바이트 크기의 판별 정보를 구한다. ROI 모양은 사용자가 생각한 ROI가 뾰족한 모양일지라도 타원이나 사각형과 유사한 모양으로 충분히 그리는 특징을 가지고 있다. 판별 알고리즘은 웨이블릿 도메인에서의 ROI 마스크에서 코드블록 단위의 마스크 정보를 스캔하여, ROI 웨이블릿 계수의 분포량에 따라 6 가지 중에 하나의 패턴으로 판별하게 된다.

<표 1>은 본 논문에서 사용하는 6 가지 패턴이다. (그림 5)는 32\*32 코드블록 크기의 스캔 순서이다. ROI 코드블록의 판별 과정은 M(3, 3), M(3, 30), M(30, 30), M(30, 3)의 4 마스크 정보를 1차로 먼저 스캔하여, 해당 코드블록의 마스크 정보가 P<sub>0</sub>혹은 P<sub>1</sub>인지를 먼저 판별을 한다. P<sub>0</sub>혹은 P<sub>1</sub>이 아니면, 2차로 A, B, C, D 방향으로 스캔을 하여 해당 코드블록의 마스크 정보와 가장 유사한 패턴을 판별한다.

A 방향의 스캔 순서는 M(3, 4), M(3, 5), ... , M(3, 29)이고, B 방향의 스캔 순서는 M(4, 30), M(5, 30), ... , M(29,



(그림 4) ROI와 ROI 마스크



(그림 5) 32\*32 코드블록의 스캔 순서

30)이고, C 방향의 스캔 순서는 M(30, 29), M(30, 28), ... , M(30, 4)이고, D 방향의 스캔 순서는 M(29, 3), M(28, 3), ... , M(4, 3)이다. 하나의 ROI 코드블록의 판별 과정은 다음과 같다.

- ① ROI 마스크 정보로부터 코드블록 단위의 마스크 정보를 가져온다.
- ② 1), 2), 3), 4) 점을 스캔하여  $P_0$  혹은  $P_1$  인지를 판별한다.
- ③ 위에서 판별되지 않으면, A, B, C, D 방향으로 스캔을 하여, ROI 둘레와 ROI와 BG 웨이블릿 계수가 접하는 a, b, c, d 좌표를 구한다. 여기서 a 좌표는 A 방향으로 스캔하여 초기 마스크 정보(M(3, 3))와 다른 마스크 정보가 처음으로 나타나는 위치를 의미한다.
- ④ ROI 둘레가 배경 임계값보다 작으면,  $P_0$ 로 판별된다.
- ⑤ ROI 둘레가 ROI 임계값보다 크면,  $P_1$ 로 판별된다.
- ⑥ 위에서 판별되지 않으면, a, b, c, d 좌표를 이용하여 기울기를 구한다.
- ⑦ 기울기와 1), 2), 3), 4) 좌표에 의해  $P_{2-5}$  중에 하나의 패턴으로 판별한다.
- ⑧ 마지막으로 a, b, c, d 좌표에 의해 폭을 계산한다.

ROI 둘레는 ROI 코드블록 내에 ROI 웨이블릿 계수의 포함률을 의미한다. 코드블록 판별 정보는 패턴(3 비트)과 폭(5 비트) 정보로 구성된다. 패턴 정보는 <표 1>의 각 패턴 번호를 의미한다. 폭 정보는 ROI 코드블록 내의 ROI 웨이블릿 계수가 차지하는 양을 의미하며, 32\*32 코드블록인 경

우는 5 비트로 표현을 하고, 64\*64 코드블록인 경우는 폭 정보를 2로 나누어서 표현한다. 이와 같은 방법으로 모든 코드블록의 판별 과정이 끝나면 각 코드블록마다 1 바이트로 구성된 ROI 판별 정보를 ROI 코딩을 위해 인코더로 전송한다.

### 3.2 코드블록의 우선적 처리

인코더에서는 디코더로부터 수신된 코드블록 판별 정보를 기반으로 우선적 처리를 한다. 우선적 처리 방법은  $P_1$ 인 경우는 코드블록 단위로 우선적 처리하고,  $P_2, P_3, P_4, P_5$ 인 경우는 배경 웨이블릿 계수의 우선권 조절 과정을 거친 후에 코드블록 단위로 우선적 처리를 한다. 다음은 제안한 ROI 코딩 방법에서의 우선적 처리 알고리즘이다. 우선적 처리 과정은 해당 코드블록이 LL 밴드에 속하지 않으면서 패턴 번호가  $P_2, P_3, P_4, P_5$ 인 경우에는 해당 코드블록 내의 배경 웨이블릿 계수에 대한 우선권 조절 처리를 거친 후, EBCOT의 Tier-1 과정을 통해서 다시 코딩을 한다. 그런 후에 해당 코드블록의 외곽에 ROI 가중치를 곱하여 ROI 코드블록 우선적 처리를 한다. 그리고  $P_1$ 인 경우는 BG 우선권 조절 처리를 거치지 않고 바로 해당 코드블록의 외곽에 ROI 가중치를 곱하여 우선적 처리를 한다.

관심영역 코드블록 내의 배경 웨이블릿 계수에 대한 우선권 조절 방법은 배경 웨이블릿 계수의 k LSB만큼 절단하는 방법으로 실현된다. (식 1)은 관심영역 처리 전의 관심영역 코드블록 내의 웨이블릿 계수인  $a(u, v)$ 을 우선권 조절 후의 웨이블릿 계수인  $\bar{a}(u, v)$ 로 변환하는 함수이다. (식 1)에서 k는 배경 웨이블릿 계수의 우선권 조절 변수이다.

$$\bar{a}(u, v) = \begin{cases} \text{sign}[a(u, v)] \left\lfloor \frac{|a(u, v)|}{2^k} \right\rfloor 2^k, & \text{BG 계수} \\ a(u, v), & \text{ROI 계수} \end{cases} \quad (\text{식 1})$$

EBCOT에서 각 품질 레이어는 코드블록(혹은 패킷, 프리시드)의 임베디드 비트 스트림으로부터 임의의 공헌도를 포함한다. 따라서 코드블록 우선적 처리는 각 코딩 패스에서 손실율을 조절한 후, PCRD(post-compression rate-distortion) 최적화 알고리즘을 다시 수행한다. 전체 손실 최소화에 의해서 코드블록 공헌도를 할당하기 때문에, 관심영역은 손실 감소와 관심영역이 일치하도록 코드블록 공헌도를 할당한다.

(식 2)는 손실 계산 방법을 나타낸다.

$$D_i^{n_i} = \begin{cases} W_{ROI} w_b \sum_{u, v \in B_j} (\hat{a}^{n_i}(u, v) - a(u, v))^2, & \text{ROI 코드블록} \\ w_b \sum_{u, v \in B_j} (\hat{a}^{n_i}(u, v) - a(u, v))^2, & \text{그외} \end{cases} \quad (\text{식 2})$$

(식 2)에서  $W_{ROI}$ 는 관심영역 가중치이며 관심영역 중요도에 따라 값을 조절할 수 있다.  $D_i^{n_i}$ 는  $n_i$ 에서의 가중 MSE 손실이고,  $w_b$ 은  $B_j$ 를 포함하는 서브밴드의 가중치이고,  $B_j$ 는

```

1: code LL band and send the resulting bit-stream to decoder
   {
2:   accept co-ordinates of user-defined ROI
3:   generate ROI mask in the wavelet domain
4:   identify ROI code blocks based on Ppatterns
   }

5: for each CB
   {
6:   Pnum = GetPatternNum( CB )
7:   if Pnum > 1 AND CB ∉ LL subband {
8:     ReduceBGPriority( Pnum, GetWwidth( CB ) )
9:     for bitplane = MSB : LSB
       {
10:      multipass coding {
11:        encode each pass
12:        calculate coding pass rate R
13:        calculate coding pass distortion estimate D
       } //end
      } //end bitplane
   }
14:   if Pnum > 0
15:     D = D * Wroi
   } //end CB
16: rate control to pack bit stream into quality-layered structure
    
```

$j$ 번째 코드블록이고,  $a(u, v)$ 는 웨이블릿 계수이고,  $\hat{a}(u, v)$ 는  $n_i$ 으로 양자화된 웨이블릿 계수이고,  $n_i$ 는 절단점(truncation point)을 의미한다.

#### 4. 실험 및 평가

실험은 관심영역 코딩을 하지 않은 EBCOT와 Maxshift 방법, 목시적 방법, 수정된 목시적 방법, 제안한 방법을 사용한다. 동일한 실험 조건을 위해 목시적 방법, 수정된 목시적 방법, 제안한 방법에서  $W_{ROI}$ 의 값은 4096로 설정하였으며, 수정된 목시적 방법과 제안한 방법에서  $k$ 의 값은 5로 설정하였다. 제안한 방법에서 관심영역 코드블록 임계값은 90, 배경 코드블록 임계값은 20으로 설정하였다. 관심영역 임계값이 90이면, 코드블록 내에 관심영역 경계면이 직각일 때는 약 90%의 관심영역을 포함하고, 대각선일 때는 약 84%의 관심영역을 포함한다. 배경 임계값이 20이면, 코드블록 내에 관심영역 경계면이 직각일 때는 약 10%의 관심영역을 포함하고, 대각선일 때는 약 8%의 관심영역을 포함한다.

관심영역의 모양은 모두 사각형으로 하였으며, 관심영역의 위치는 이미지의 중앙 부분으로 하였다. 관심영역의 크기는 전체 이미지의 25%이다. 실험 이미지는 <표 2>와 같으며, 모두 흑백 이미지(8 bpp)이다.

<표 2> 실험 이미지

파일명	해상도	종류
Lenna	512 * 512	자연 이미지
Ultrasound	500 * 370	초음파 이미지

#### 4.1 주관적 화질 평가

(그림 6)은 0.25bpp에서 다양한 코딩 방법을 사용하여 재구성된 이미지들이다. a)는 원 이미지를 나타내고, b)는 관심영역 코딩을 하지 않은 EBCOT 방법으로서 전체 이미지가 희미하게 보이고, c)는 Maxshift 방법으로서 관심영역 부분만 고 품질로 보이고, d)와 e)는 수정된 목시적 방법과 제안한 방법으로서 고품질의 관심영역과 저품질의 배경을 보이고 있으며, 두 방법은 눈으로 봤을 때 큰 차이가 없다. (그림 7)은 제안한 방법을 이용하여 0.125, 0.5, 1.0, 2.0bpp로 재구성된 Lenna 이미지이다.

#### 4.2 객관적 화질 평가

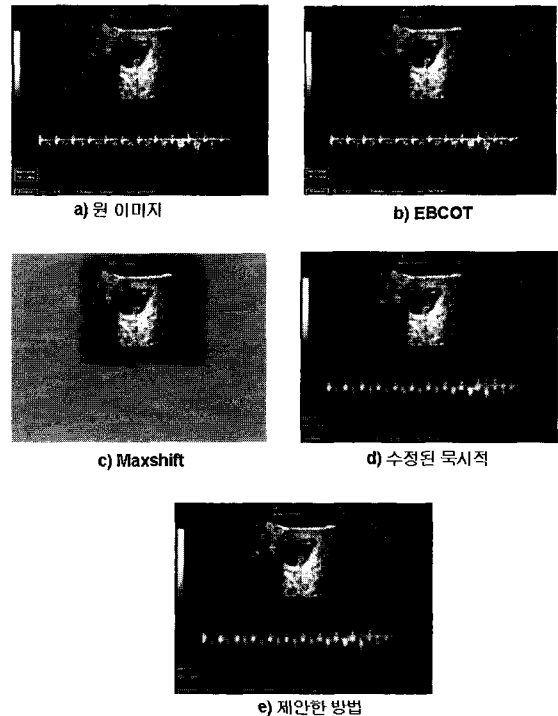
본 논문에서는 PSNR를 사용하여 객관적 화질 평가를 하였다. 샘플당  $n$  비트의 길이를 가지는 이미지를 위한 PSNR은 (식 3)이고,

$$PSNR = 10 \log_{10} \frac{(2^n - 1)^2}{MSE} \quad (식 3)$$

MSE를 구하는 함수는 (식 4)과 같다.

$$MSE = \frac{\sum (\hat{x} - x)^2}{A} \quad (식 4)$$

여기서  $x$ 는 원 화소 값을 의미하고,  $\hat{x}$ 는 재구성된 화소 값을 의미하고,  $A$ 는 관심영역 영역을 의미하고, 따라서 관



(그림 6) 0.25bpp에서 다양한 방법으로 재구성된 "Ultrasound" 이미지



(그림 7) 제안된 방법으로 재구성된 "Lenna" 이미지

<표 3> 다양한 방법에 대한 ROI PSNR 비교

코딩방법 비트율	EBCOT (ROI X)	목시적	Maxshift	수정된 목시적	제안한 방법
0.0625	23.41	24.20	27.89	24.52	24.67
0.125	25.06	27.30	30.29	28.11	28.38
0.25	27.80	30.92	33.88	32.32	32.62
0.5	31.62	35.90	39.01	37.92	38.29
1.0	37.87	43.23	47.73	45.98	46.31

심영역 PSNR은 관심영역의 R-D 성능을 측정하여 구할 수 있다. <표 3>은 초음파 이미지에 대한 각 비트율에서 관심영역의 PSNR값을 나타내고 있다.

관심영역의 크기는 (그림 6)과 같이 전체 이미지의 25%이다. <표 3>과 같이 수정된 목시적과 제안한 방법의 PSNR 값이 큰 차이가 없다. <표 4>은 무손실 관심영역 재구성을 위해 요구되는 비트율을 나타낸다.

관심영역의 크기가 전체 이미지의 25%인 경우 관심영역의 완전한 재구성은 EBCOT는 2.931bpp, 목시적 방법은 1.846bpp, 수정된 목시적 방법은 1.407bpp에서 이루어지며, 제안한 방법은 1.421bpp에서 이루어져 수정된 목시적 방법과 큰 차이가 없다.

<표 4>와 같이 관심영역의 크기가 작을수록 관심영역의 무손실 재구성 비트율도 낮아진다. 제안한 방법과 수정된 목시적 방법은 저비트율에서 전체 이미지에 대한 압축 성능이 뛰어났다. 실험 이미지 외의 다른 이미지에서도 비슷한 결과를 관찰할 수 있다. 하지만 제안한 방법도 수정된 목시적 방법과 마찬가지로 관심영역 코드블록 내의 배경 웨이블릿 계수를 삭제하기 때문에 JPEG2000 표준 관심영역의 무손실 코딩과는 호환이 되지 않지만, 손실 코딩과는 호환이 된다.

<표 4> 무손실 관심영역 재구성을 위해 요구되는 비트율

ROI 코딩 방법	비트율	
	25% ROI	8% ROI
EBCOT	2.931	2.931
목시적	1.846	1.106
Maxshift	1.317	0.526
수정된 목시적	1.407	0.646
제안한 방법	1.421	0.652

#### 4.3 관심영역의 코딩 시간 비교

동적 관심영역은 디코딩 과정에서 관심영역 코딩을 하여 복원을 하기 때문에, 빠른 복원을 위해서는 빠른 관심영역 코딩이 요구된다. 관심영역 코딩 시간은 사용자의 관심영역 모양 정의 시간, 관심영역 코드블록 판별시간, 우선적 처리 시간의 합이다. 여기서 사용자의 관심영역 모양 정의 시간은 동일하다고 가정하며, 우선적 처리 방법은 모두 동일하다. 결국 관심영역 코딩 시간은 관심영역 코드블록 판별 시간에 의해 결정된다. 기존의 두 방법은 판별을 위해 순차 스캔을 하는 반면에, 제안한 방법은 (그림 5)와 같이 일부 웨이블릿 계수만 스캔한다. <표 5>는 관심영역 크기에 따른 코드블록당 평균 스캔 횟수 비교이다.

배경 코드블록 판별을 위해 목시적과 수정된 목시적 방법은 모든 코드블록 마스크를 스캔해야 알 수 있지만, 제안한 방법에서는 4번의 스캔과정을 통해 배경 코드블록을 판별한다. 관심영역 코드블록 판별을 위해 목시적 방법은 한 번의 스캔으로 판별을 하고, 수정된 목시적 방법은 한 번의 스캔으로 판별한 후에, 관심영역 코드블록 내의 배경 웨이블릿 계수에 대한 우선권 조절을 위해 모든 코드블록 마스크 정보를 스캔한다. 반면에 제안한 방법은 4번의 스캔과정을 통해 알 수 있다.

배경 웨이블릿 계수를 포함하는 관심영역 코드블록 판별을 위해 목시적 방법은 계산의 편리를 위해 전체 스캔 수의 반으로 하고, 수정된 목시적 방법에서는 관심영역 코드블록 내의 배경 웨이블릿 계수에 대한 우선권 조절을 위해 모든 코드블록의 마스크 정보를 스캔한다. 제안한 방법은 평균적으로 4 + (n-6) 번의 스캔과정을 통해 알 수 있다. 만약 관심영역 크기가 전체 이미지의 25%인 경우, 배경 코드블록은

<표 5> 관심영역 크기에 따른 코드블록당 평균 스캔 횟수 비교  
n : 코드블록의 가로크기

코딩방법 스캔횟수		목시적	수정된 목시적	제안한 방법
배경 CB		$n^2$	$n^2$	4
ROI CB		1	$n^2$	4
배경WC를 포함 하는 ROI CB		$n^2/2$	$n^2$	4+(n-6)
ROI 크기	25% ROI	$0.75n^2+0.2$	$(0.7+0.2+0.1)n^2$	$0.1n+3.4$
	8% ROI	$0.915n^2+0.05$	$(0.88+0.05+0.07)n^2$	$0.07n+3.58$

70%, 관심영역 코드블록은 20%, 배경 웨이블릿 계수를 포함한 관심영역 코드블록은 10% 정도 차지한다. 따라서 코드블록의 가로 크기가  $n$ 이라면, 3 가지 방법의 코드블록당 평균 스캔 수는  $0.75n^2 + 0.2, n^2, 0.1n + 3.4$ 이다. 만약 관심영역 크기가 전체 이미지의 8%인 경우, 배경 코드블록은 88%, 관심영역 코드블록은 5%, 배경 웨이블릿 계수를 포함한 관심영역 코드블록은 7% 정도 차지한다. 따라서 코드블록의 가로 크기가  $n$ 이라면, 3 가지 방법의 코드블록당 평균 스캔 수는  $0.915n^2 + 0.05, n^2, 0.07n + 3.58$ 이다.

4.4 관심영역 코딩 방법 비교

<표 6>는 관심영역 코딩 방법 비교이다.

<표 6> 관심영역 코딩 방법 비교

관심영역 코딩 방법	①Maxshift	②묵시적	③수정된 묵시적	④제한한 방법
	정적	동적		
우선적 처리단위	WC	CB		
관심영역 정확도	○	△	○	□
무손실로 재구성	○	○	×	□
관심영역 중요도조절	×	○	○	○
관심영역 코딩속도	의미없음	□	△	○
Published Year	2000	2002	2005	2007

○:고 □:중 △:저 ×:지원안함

①은 정적 관심영역으로서 웨이블릿 계수 단위로 우선적 처리하고, ②, ③, ④는 동적 관심영역으로서 코드블록 단위로 우선적 처리를 한다. 관심영역 정확도에 있어서는 ①이 웨이블릿 계수 단위로 우선적 처리를 하기 때문에 가장 우수하며, 다음으로 ③, ④, ② 순으로 우수하다. 무손실 재구성은 ①과 ②는 가능하며, ③은 관심영역 코드블록 내의 모든 배경 손실 처리하고, ④는 ③을 개선하여 배경 손실량을 줄였다. 관심영역 중요도 조절은 ①은 불가능한 반면에, ②, ③, ④는 모두 가능하다. 관심영역 코딩 속도는 ①은 정적 관심영역 방법이기 때문에 큰 의미가 없고, ④가 가장 빠르며, 다음으로 ②, ③순으로 빠르다.

5. 결론 및 향후과제

본 논문에서는 압축된 비트 스트림으로부터 사용자 위주의 관심영역을 추출하여 빠르게 관심영역 코딩한 후에 이미지를 전송하여 복원하는 동적 관심영역 코딩 방법을 다루고자 하였다. 기존 동적 ROI 코딩 방법은 관심영역 코딩을 위해 많은 스캔 과정을 필요로 하고, 관심영역 경계 영역에 있는 배경을 우선적 처리 문제와 손실 문제로 인해, 정확하고 빠른 코딩이 이루어지지 못하였다. 본 논문에서는 이러

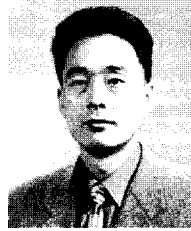
한 문제점들을 개선하기 위하여 관심영역 임계값과 배경 임계값을 이용한 패턴 기반의 코드블록 판별과 관심영역 코드블록 내의 배경 웨이블릿 계수에 대한 우선권 조절을 일괄 처리함으로써 빠른 관심영역 코딩이 가능하였다. 제안한 방법의 유용성을 보이기 위해 기존의 방법들과 동일한 매개변수 값을 설정하여 다양한 비트율에서 비교 실험을 하였고, 각 방법들에 대한 코딩 시간도 비교하였다. 실험 결과, 제안한 방법은 기존의 방법들과 성능 면에 있어서는 크게 차이가 없으면서도 코딩 시간을 단축함으로써 빠른 이미지 전송이 가능함을 확인하였다. 향후 과제로는 실제로 사용자가 관심영역을 정의한 이후부터 관심영역 코딩, 전송, 복원까지의 시간을 비교 실험하는 연구와 관심영역 임계값과 배경 임계값에 따른 실험하는 연구가 있다.

참고 문헌

- [1] M. Rabbani and R. Ioshi, "An Overview of the JPEG2000 Still Image Compression Standard," *Signal Processing : Image Communication*, Vol.17, pp.3-48, 2002.
- [2] 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.
- [3] Martin Boliek and Charilas Christopoulos, "JPEG 2000 Part I Final Committee Draft Version 1.0," *ISO/IEC JTC 1/SC 29/WG 1 N1646R*, Mar., 2000.
- [4] Martin Boliek, Eric Majani, J. Scott Houchin, James Kasner and Mathias Larsson Carlander, "JPEG 2000 Part II Final Committee Draft," *ISO/IEC JTC 1/SC 29/WG 1 N2000*, Dec., 2000.
- [5] 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.
- [6] Kong H-S, Vetro A., Hata T. and Kuwahara N., "Fast Region-of-Interest Transcoding for JPEG2000 Images," *Mitsubishi Electric Reserch Laboratories, Inc.*, Dec., 2005.
- [7] Andrew P. Bradley and Fred W. M. Stentiford, "JPEG 2000 and Region of Interest Coding," *Digital Image Computing Techniques and Applications*, Jan., 2002.
- [8] R.Grosbois, D.Santa-Crus and T.Ebrahimi, "New approach to JPEG2000 compliant Region of Interest coding," *Proc. of the SPIE 46th*, July 29th Aug., 3rd, 2001.
- [9] Lijie Liu and Guoliang Fan, "A New JPEG2000 Region-of-Interest Image Coding Method : Partial Significant Bitplanes Shift," *IEEE Signal Processing Letters*, Vol. 10, No. 2, Feb., 2003.
- [10] Rene Rosenbaum and Heidrun Schumanm, "Flexible, Dynamic and Compliant Region of Interest coding in JPEG2000," *IEEE ICIP*, pp.22-25, Sep., 2002.
- [11] Victor Sanchez, Anup Basu, and Mrinal K. Mandal,

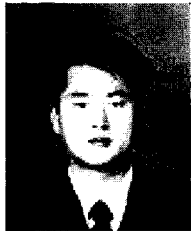
"Prioritized region of interest coding in JPEG2000," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 14, No. 9, Sep., 2004.

### 유 창 열



e-mail : yoocyeul@nate.com  
1987년 경상대학교 전산통계학과 학사  
1993년 경상대학교 대학원 전자계산학과  
공학석사  
1999년 경상대학교 대학원 전자계산학과  
공학박사  
1996년 3월~현재 남해전문대학 컴퓨터응용  
정보과 부교수

관심분야 : 시스템분석, ERP 성능평가, 컨설팅 등

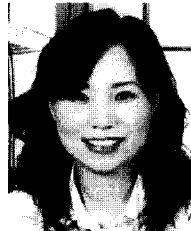


### 강 기 준

e-mail : kijuni@gnu.kr  
2000년 경상대학교 컴퓨터학과 학사  
2002년 경상대학교 교육대학원  
컴퓨터교육과 석사  
2007년 경상대학교 대학원 컴퓨터학과  
박사

2006년~현재 거상테크놀로지 근무  
관심분야 : JPEG2000, MPEG, 원격교육

### 박 순 화



e-mail : cpr8282@nate.com  
1996년 진주 산업대학교 전자계산학과  
학사  
2000년 경상대학교 대학원 컴퓨터학과  
석사  
2005년 경상대학교 대학원 컴퓨터학과  
박사과정 수료

관심분야 : Motion-JPEG2000, 멀티미디어 통신



### 서 영 건

e-mail : young@ggnu.ac.kr  
1987년 경상대학교 전산통계학과 학사  
1989년 숭실대학교 대학원 전자계산학과  
석사  
1997년 숭실대학교 대학원 전자계산학과  
박사

1989년~1992년 삼보컴퓨터 근무  
1997년~현재 경상대학교 컴퓨터교육과 부교수  
2001년~현재 경상대학교 컴퓨터정보통신연구소원  
관심분야 : 멀티미디어통신, 영상인식, 원격교육

### 이 점 숙



e-mail : alleya@empal.com  
1994년 경상대학교 전자계산학과 학사  
1999년 경상대학교 교육학과(전산교육)  
석사  
2005년 경상대학교 컴퓨터학과  
박사과정 수료

관심분야 : Wavelet, JPEG2000, Software  
Process



### 박 재 흥

e-mail : pjh@gnu.ac.kr  
1978년 충북대학교 수학교육과 학사  
1980년 중앙대학교 전산학과 석사  
1989년 중앙대학교 전산학과 박사  
1983년~현재 경상대학교 컴퓨터학과 정교수

관심분야 : Software Testing, Software  
Process

### 이 부 권



e-mail : bklee@ggnu.ac.kr  
1972년 경상대학교 농경제학과 학사  
1978년 미시건주립대학 대학원 시스템  
공학전공 석사  
2001년 경남대학교 대학원 컴퓨터공학과  
박사

1980년~현재 경상대학교 컴퓨터학과 교수  
2001년~현재 경상대학교 컴퓨터정보통신연구소원  
관심분야 : 시뮬레이션, 멀티미디어, 원격교육