

동적 관심영역 코딩을 위한 효율적인 관심영역 코드블록 판별 알고리듬

강기준[†], 안병태^{††}

요 약

본 논문에서는 동적 관심영역 코딩을 위한 효율적인 관심영역 코드블록 판별 알고리듬을 제안한다. 제안한 알고리듬은 관심영역 코드블록 판별 시간을 줄이기 위하여 관심영역 모양의 특징을 고려하여 일부 마스크 정보만으로 관심영역의 포함율을 계산하고, 포함율과 관심영역 임계값에 의해 관심영역 코드블록 유무를 판별한다. 그리고 판별 알고리듬은 관심영역 임계값을 조절함으로서 관심영역 코드블록 내의 배경 웨이블릿 계수의 우선적 처리와 손실 부분을 조절도 할 수 있었다. 제안한 방법의 유효성을 나타내기 위해 기존의 방법들과 비교 실험을 통해 제안한 방법이 기존의 방법에 비해 품질과 속도 면에 있어서 우수함을 확인하였다.

An Efficient ROI Code Block Discrimination Algorithm for Dynamic ROI Coding

Ki-Jun Kang[†], Byeong-Tae Ahn^{††}

ABSTRACT

This paper proposes an efficient ROI code block discrimination algorithm for dynamic ROI coding. The proposed algorithm calculates the girth of the ROI only with some mask information in consideration of the characteristics of the shape of the ROI for reducing a ROI code block discrimination time, and this proposed algorithm discriminates whether there is a ROI code block by the girth and the critical value of the ROI. Also, this discrimination algorithm is capable of treating the coefficients of the background within a ROI code block preferentially and controlling a loss by controlling the threshold value of the ROI. In order to demonstrate the utility of the proposed method, this paper conducted a comparative experiment of the proposed method with the existing methods. As a result of this experiment, it was confirmed that the proposed method was superior to the conventional methods in terms of quality and speed.

Key words: JPEG2000, Region-Of-Interest(관심영역), EBCOT(Embedded Block Coding with Optimized Algorithm)(EBCOT 알고리듬)

1. 서 론

영상 정보는 인터넷과 멀티미디어 기술의 발전으로 의료진단, 웹 브라우징, 이미지 데이터베이스 그

리고 컴퓨터 통신 등의 광범위한 응용분야에서 활용되고 있다[1,2]. 하지만 대용량의 영상 정보를 저장 및 전송하기 위해서 많은 메모리와 넓은 대역폭을 요구하기 때문에 신속한 영상 처리가 어려운 실정이

* 교신저자(Corresponding Author) : 안병태, 주소 : 경기도 부천시 소사구 괴안동 185-34 유한대학 경영정보과 (422-749), 전화 : 02)2610-0683, FAX : 055)762-1944,
E-mail : ahnbnt@hanmail.net

접수일 : 2007년 7월 16일, 완료일 : 2007년 10월 30일

[†] 준회원, 거상 테크놀로지
(E-mail : kijuni@gnu.kr)

^{††} 정회원, 유한대학 경영정보학과

다. 이를 해결하기 위하여 전체 영상을 전송 및 복원 하기보다는 영상의 일부 특정 영역이나 사용자 위주의 관심영역(Region of interest)에 대한 우선적 처리가 요구되고 있다. 이러한 우선적 처리는 JPEG2000에서 표준 관심영역 코딩 방법들을 비롯하여, 이를 보완한 새로운 관심영역 코딩 방법들에 대한 연구가 최근까지 활발하게 이루어지고 있다[3]. 관심영역 코딩 방법은 언제 관심영역을 지정하느냐에 따라서 정적 관심영역과 동적 관심영역 코딩 방법으로 나뉜다. 정적 관심영역 코딩 방법은 압축 과정에서 관심영역 모양을 지정하여 관심영역 코딩하는 방법으로 대부분의 응용에서 적용이 가능하다. 동적 관심영역 방법은 복원 과정에서 사용자 위주의 관심영역 모양을 지정하여 관심영역을 코딩하는 방법으로서 관심영역 코딩되어 있지 않은 압축된 비트스트림으로부터 임의의 관심영역을 추출하여 우선적 처리를 하는 방법이다[4]. 이 방법에서는 표준 방법인 뮤시적 방법과 이를 보완 수정한 뮤시적 방법이 있다. 하지만 기존의 동적 관심영역 코딩 방법은 관심영역 코드블록(Code Block) 내의 배경 웨이블릿 계수(Wavelet Coefficient)에 대한 우선적 처리 문제와 손실 문제 그리고 관심영역 코드블록을 판별하기 위한 수많은 스캔과정을 필요로 하여 코딩 속도를 저하시키는 문제가 발생하였다. 본 논문에서는 이러한 문제점들을 보완한 효율적인 관심영역 코드블록 판별 알고리듬을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서는 관련연구로 JPEG2000 압축 알고리듬과 관심영역 코딩 방법에 대해 알아본다. 3장에서는 본 논문에서 제안하는 알고리듬과 이를 적용한 관심영역 코딩 방법을 설명한다. 4장에서는 실험을 통하여 제안한 방법의 유효성을 검증한다. 마지막으로 5장에서는 결론 및 향후 연구 과제를 제시한다.

2. 관련연구

2.1 JPEG2000 이미지 압축 표준

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

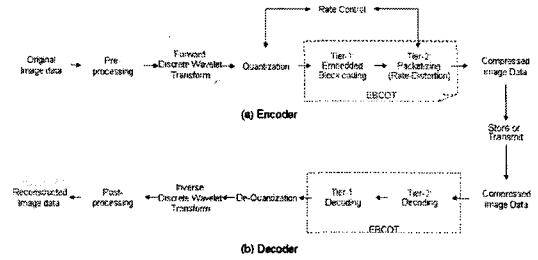


그림 1. JPEG2000 블록 다이어그램

JPEG2000의 대표적인 특성은 무손실/손실 압축, 무손실 코딩에서의 내포된 손실, 화소 정확성이나 해상도에 의한 진보적인 전달, 비트 오류 그리고 관심영역 코딩 등이 있다. JPEG2000의 필요성이 강조되는 다양한 응용으로는 인터넷, 컬러 팩스, 인쇄, 스캐닝, 디지털 사진, 이동식 적용, 의학 이미지, 전자 도서관과 전자상거래 등이 있다[2,5].

그림 1의 (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 역순으로 한다.

2.2 관심영역 코딩 방법

관심영역 코딩 방법은 JPEG2000의 주요 특징 중의 하나로서 이미지 내의 관심영역을 배경보다 더 나은 품질로 빠르게 처리하는 기술로서 이미지 내의 특정 영역이 다른 영역보다 높은 중요성을 가지는 경우에 적용할 수 있다. 관심영역 이미지의 장점은 점진적 이미지 전송 방법보다 압축률과 전송시간을 단축시킬 수 있을 뿐만 아니라 효율적인 메모리 관리 등으로 인해 사용자의 다양한 요구에 응할 수 있다 [6,7]. 관심영역 코딩 방법은 크게 정적 관심영역 코딩 방법과 동적 관심영역 코딩 방법으로 나뉜다. 정적 관심영역 코딩 방법은 인코딩 과정에서 양자화된 웨이블릿 계수들 중에 사용자가 지정한 관심영역 모양에 의해 관심영역 웨이블릿 계수들을 판별하여, 관심영역 중요도에 따라 관심영역 웨이블릿 계수들의 비트-평면(Bit-Plane)을 이동시키는 방법이다. 이 방법은 한번 관심영역 코딩을 하게 되면 관심영역 모양과 중요도를 변경할 수 없다. 이 방법에는

Maxshift[7], Scaling based[7], Maxshift-like[8], PSBS hift[9] 등이 있다. 인코딩 과정에서 관심영역을 미리 알지 못하는 경우에 이 방법을 사용할 수 없다. 이러한 경우에는 처음 디코딩 과정 중에서 사용자가 원하는 관심영역 모양 정보를 인코더 측에게 넘겨주면 인코더 측에서는 그 정보를 토대로 관심영역 코딩이 되어 있지 않은 나머지 압축된 비트스트림으로부터 임의의 관심영역을 판별하여 우선 처리를 하게 한다. 이렇게 관심영역 코딩을 실현하는 방법을 동적 관심영역 코딩 방법이라고 한다. 대표적인 동적 관심영역 코딩 방법에는 묵시적(Implicit)[7], 수정된 묵시적(Modified Implicit)[10], Flexible and dynamic[11], Fast ROI Transcoding[4], Prioritized[12] 등이 있다.

묵시적 방법은 Part 1 표준으로서 코드블록 내에 관심영역 웨이블릿 계수가 하나 이상 포함되면 관심영역 코드블록으로 판별되어 우선적으로 처리한다. 이로 인해 관심영역 코드블록에 포함된 배경 웨이블릿 계수도 우선적 처리를 하는 문제가 발생한다. 이를 보완하기 위해 수정된 묵시적 방법은 관심영역 코드블록에 포함된 배경 웨이블릿 계수들의 우선권을 줄이기 위해 관심영역 코드블록에 포함된 배경 웨이블릿 계수들의 비트-평면 중에서 k 만큼의 LSB(Least Significant Bit-planes)를 삭제한 후에 관심영역 코드블록으로 우선적 처리한다.

2.3 기존 관심영역 코딩 방법의 문제점

동적 관심영역 코딩 방법은 사용자에 의해 정의된 관심영역 정보를 토대로 관심영역을 코딩한 후에 이미지를 복원하기 때문에 저비트율에서 고품질의 관심영역을 서비스하기 위하여 무엇보다도 빠른 관심영역 코딩이 요구된다. 하지만 묵시적 관심영역에서는 배경 웨이블릿 계수를 포함하는 관심영역 코드블록을 우선적으로 처리함으로써 부정확한 관심영역 코딩, 관심영역 품질 저하 및 관심영역 코드블록 판별에 따른 많은 시간이 소요되는 단점이 발생한다.

수정된 묵시적 관심영역에서는 관심영역 코드블록 내의 배경 웨이블릿 계수에 대한 우선권 조절을 통해 정확한 관심영역 코딩과 관심영역 품질을 개선 시켰지만 우선권 조절에 대한 계산복잡도로 인해 관심영역 코딩 속도를 저하시키는 문제가 발생한다. 또

한 배경 웨이블릿 계수에 대한 우선권을 조절하여도 여전히 배경 웨이블릿 계수의 우선적 처리 문제가 남아있다. 이러한 여러 가지 문제들은 관심영역 코드블록 내의 배경 웨이블릿 계수의 수가 많을수록 커지게 된다.

3 개선된 동적 관심영역 코딩 방법

그림 2는 제안한 방법의 전체 구성도이다. 인코더와 디코더는 하나의 컴퓨터 시스템에 함께 존재할 수도 있고, 네트워크로 연결된 별도의 컴퓨터 시스템에 각각 존재할 수도 있다. 인코더와 디코더가 별도의 컴퓨터 시스템에 존재하는 경우에는 관심영역 마스크 생성과 관심영역 코드블록 판별 과정을 디코더에서 처리하여 적은 양의 관심영역 판별 정보를 인코더로 보낼 수 있어 보다 나은 처리결과를 얻을 수 있다.

전체 구성을 살펴보면, (1) 관심영역을 결정하기 위해 관심영역 코딩이 되어 있지 않은 압축된 비트스트림으로부터 기본적인 인코딩 정보와 LL 밴드 정보를 추출하여 디코더로 전송한다. (2) 인코더로부터 수신된 LL 밴드를 복원하고, 사용자는 대략적인 이미지를 보고 관심영역 모양을 지정한다. 그래서 이미지 도메인에서의 관심영역 마스크를 생성한다. (3) 이미지 도메인에서의 관심영역 마스크, 기본적인 인코딩 정보, IDWT(Inverse DWT) 식을 이용하여 웨이블릿 도메인에서의 관심영역 마스크를 생성한다. (4) 본 논문에서는 제안한 관심영역의 모양 특징을 고려한 효율적인 마스크 스캔 과정을 통해, 관심영역의 포함율을 계산한다. (5) 관심영역의 포함율이 관

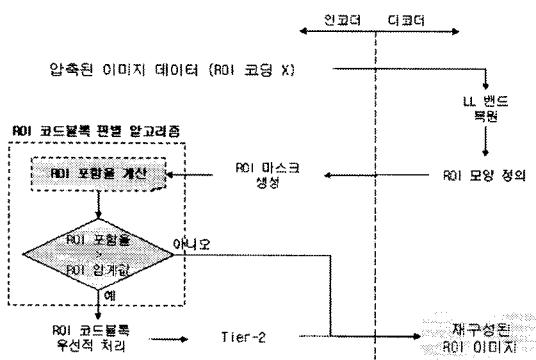


그림 2. 제안한 방법의 전체 구성도

심영역 임계값보다 크면 관심영역 코드블록으로 판별한다. (6) 관심영역 코드블록은 중요도에 따라 관심영역 가중치를 할당한 후에 Tier-2 과정을 거쳐 우선적인 처리를 한다.

3.1 관심영역 마스크 생성

사용자는 디코더에서 재구성된 LL밴드 정보, 즉 대량적인 이미지 정보를 보고 관심영역의 모양을 정의하게 되고, 이를 통해 이미지 도메인에서의 관심영역 마스크를 생성한다. 식(1)은 이미지 도메인에서의 관심영역 마스크 함수이다.

$$M(x, y) = \begin{cases} 1, & \text{관심영역} \\ 0, & \text{배경} \end{cases} \quad (1)$$

$M(x, y)$ 은 이진 마스크를 의미하며, 이 값이 1이면 그 위치는 관심영역 샘플을 의미하고, 이 값이 0이면 그 위치가 배경 샘플을 의미한다. 이렇게 하여 이미지 크기의 이진 마스크를 생성한다. 관심영역 모양은 다양한 모양으로 정의 될 수 있다. 관심영역 코딩을 위해서는 이미 압축된 비트 스트림으로부터 관심영역 부분을 찾는 과정, 즉 웨이블릿 도메인에서의 관심영역 마스크를 생성하는 과정이 필요하다. 이 정보는 이미지 도메인에서의 관심영역 마스크와 IDWT를 이용하면 구할 수 있다. 최종 관심영역 마스크 생성 과정은 먼저 이미지 도메인에서의 관심영역 마스크와 마지막 IDWT에 의해 2개의 서브밴드 안에 어떤 위치의 웨이블릿 계수가 관심영역 웨이블릿 계수로 재구성되기 위해 사용되었는지를 추적하게 된다. 그런 후에, 두 서브밴드에서의 관심영역 마스크와 마지막 이전의 IDWT에 의해 각각 2개의 서브밴드 안에 어떤 위치의 웨이블릿 계수가 관심영역 웨이블릿 계수로 재구성되기 위해 사용되었는지를 추적하게 된다. 이렇게 하여 각 분해레벨에서 모든 서브밴드 내에 어떤 위치의 웨이블릿 계수가 관심영역 웨이블릿 계수로 재구성하기 위해 사용되었는지를 추적하게 된다.

이와 같은 방법으로 생성된 웨이블릿 도메인에서의 관심영역 마스크는 그림 3과 같이 각 분해레벨에서 서브밴드 안에 관심영역 코드블록, 배경 코드블록 그리고 일부 배경 웨이블릿 계수를 포함하는 관심영역 코드블록들로 구성된다.

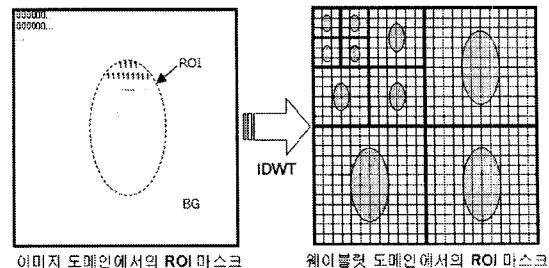


그림 3. 관심영역과 관심영역 마스크

3.2 관심영역 코드블록 판별 알고리듬

웨이블릿 도메인에서의 관심영역 마스크에서 관심영역 코드블록을 판별하는 과정이 필요하다. 본 연구에서는 기존의 방법과 같은 순차 스캔을 하지 않고, 관심영역 모양의 특징을 고려하여 효율적인 스캔 과정을 통해서 코드블록당 1 비트의 판별 정보를 구한다. 그림 4는 32×32 크기의 코드블록에서 다양한 관심영역 모양의 예제들을 나타내고 있다.

그림 4의 1과 2처럼 관심영역 크기가 코드블록보다 작은 경우는 전체 이미지에 비해 너무나 작기 때문에 고려하지 않는다. 그림 4의 3처럼 사용자가 머리속으로 생각한 관심영역이 뾰족한 모양인 경우는 사용자가 실제로 4, 5, 6과 같이 타원이나 사각형에 유사하게 관심영역을 충분하게 지정한다. 이와 같이 관심영역 모양의 특징들을 정리하면 다음과 같다.

⇒ 관심영역 크기는 32×32 보다 크다.

⇒ 경험적으로 관심영역 모양은 원, 타원, 사각형과 유사하다.

⇒ 관심영역의 중앙 부분, 관심영역과 배경의 경계 부분, 배경 부분 순으로 중요도가 높다.

⇒ 관심영역의 경계부분은 배경일 가능성이 높다.

⇒ 순수 관심영역 코드블록과 순수 배경 코드블록이 대부분을 차지한다.

⇒ 관심영역 코드블록은 인접한 둘 이상의 관심영역 경계면을 가진다.

이러한 관심영역 모양의 특징을 기반으로 마스크 정보를 스캔하여 관심영역 포함율을 계산한다. 그림 5는 32×32 코드블록 크기의 스캔 순서를 나타내고 있다.

스캔 순서는 먼저 $M(3, 3), M(3, 30), M(30, 30), M(30, 3)$ 의 4 마스크를 스캔하고, 필요하다면 A, B, C, D 방향으로 스캔한다. A 방향의 스캔 순서는

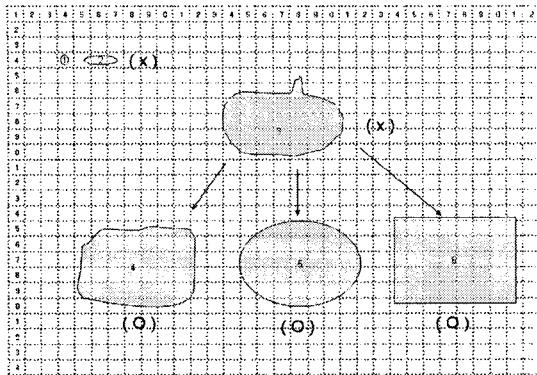


그림 4. 다양한 관심영역 모양

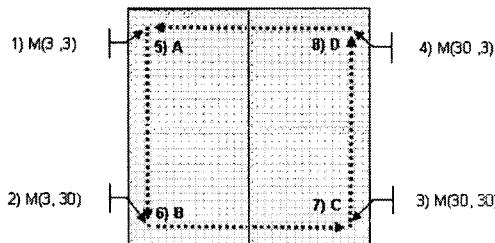


그림 5. 32×32 코드블록의 스캔 순서

$M(3, 4)$, $M(3, 5)$, ..., $M(3, 29)$ 이고, B 방향의 스캔 순서는 $M(4, 30)$, $M(5, 30)$, ..., $M(29, 30)$ 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. 다음은 관심영역 코드블록 판별 알고리듬이다.

- ① 관심영역 마스크로부터 하나의 코드블록 마스크를 가져온다.
- ② 1), 2), 3), 4)를 스캔하여 모두 1이면 순수 관심영역 코드블록으로, 0이면 순수 배경 코드블록으로 판별한다.
- ③ ②에서 판별되지 않으면, A, B, C, D 방향으로 스캔하여, 관심영역과 배경 마스크가 접하는 a, b, c, d 좌표를 구한다.
- ④ 1), 2), 3), 4)의 마스크와 a, b, c, d 좌표값을 사용하여 관심영역 둘레, 즉 포함율을 구한다.
- ⑤ 관심영역 포함율이 관심영역 임계값보다 크면, 관심영역 코드블록으로 판별되고, 그렇지 않으면 배경 코드블록으로 판별된다.

관심영역 포함율은 코드블록 내에 관심영역 웨이블릿 계수가 포함되어 있는 비율을 나타내며, 관심영역 둘레에 의해 계산된다. 관심영역 임계값은 관심영역 코딩 시에 조절이 가능하다. 관심영역 임계값이 낮을수록 관심영역과 배경의 경계에 있는 코드블록들이 관심영역 코드블록으로 판별될 가능성성이 높아지고, 결국 배경 웨이블릿 계수를 우선적 처리하는 경우가 높아 관심영역 품질이 떨어지게 된다. 본 연구에서는 실험적인 결과에 의해 기본값을 90으로 설정하였다. 일반적으로 관심영역 둘레가 90이면, 코드블록 내에 관심영역 경계면이 직각일때는 약 90%가 관심영역이고, 대각선일때는 약 84%가 관심영역이다. 그림 6은 몇 가지 코드블록 판별 예제를 나타내고 있다.

(a)는 $M(3, 3)$, $M(3, 30)$, $M(30, 30)$, $M(30, 3)$ 의 마스크 값이 모두 1이므로, 관심영역 코드블록으로 판별된다. (b) 예제는 $M(3, 3)$, $M(3, 30)$, $M(30, 30)$, $M(30, 3)$ 의 마스크 값이 동일하지 않으므로, A, B, C, D 방향으로 스캔하여 관심영역 둘레를 구한다. 둘레를 구하는 과정은 $M(3, 3)$ 과 $M(3, 30)$ 이 1로 동일하므로 A 방향의 관심영역 길이는 32가 되고, $M(3, 30)$ 과 $M(30, 30)$ 이 1로 동일하므로 B 방향의 관심영역 길이는 32가 되고, $M(30, 30)$ 과 $M(30, 3)$ 이 서로 다르기 때문에, C 방향으로 $M(30, 30)$ 과 다른 값이 나올 때까지 스캔하여 c(30, 6) 좌표를 구하고, 이 좌표에 의해 C 방향의 관심영역 길이 26($=32-6$)을 구한다. $M(30, 3)$ 과 $M(3, 3)$ 이 서로 다르기 때문에, D 방향으로 $M(30, 3)$ 과 다른 값이 나올 때까지 스캔

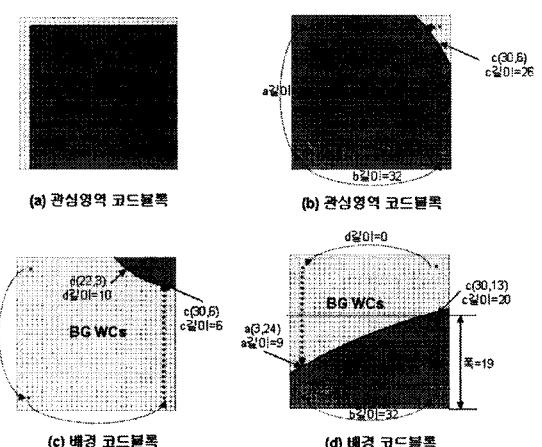


그림 6. 코드블록 판별 예제

하여 $d(27, 3)$ 좌표를 구하고, 이 좌표에 의해 D 방향의 관심영역 길이 27을 구한다. 결과적으로 관심영역 둘레는 각 방향의 관심영역 길이를 합한 117 ($=32+32+26+27$)이 된다. 이 값이 관심영역 임계값 90 보다 크기 때문에 관심영역 코드블록으로 판별된다. (c), (d)도 같은 방법을 사용하여 배경 웨이블릿 계수를 판별한다.

이와 같은 방법으로 모든 코드블록의 판별 과정이 끝나면 각 코드블록마다 1 비트로 구성된 관심영역 판별 정보는 우선적 처리를 위해 인코더로 전송한다.

3.3 코드블록의 우선적 처리

인코더에서는 디코더로부터 수신된 코드블록 판별 정보를 기반으로 우선적 처리를 한다. EBCOT에서 각 품질 레이어는 코드블록(혹은 패킷)의 임베디드 비트 스트림으로부터 임의의 공현도를 포함한다. 따라서 코드블록 우선적 처리는 각 코딩 패스에서 손실율을 조절한 후, PCRD(Post-Compression Rate-Distortion) 최적화 알고리듬을 다시 수행한다. 전체 손실 최소화에 의해서 코드블록 공현도를 할당하기 때문에, 관심영역 부분은 손실 감소와 관심영역이 일치하도록 코드블록 공현도를 할당한다. 식(2)는 손실 계산 방법을 나타낸다.

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

식(2)에서 W_{ROI} 는 관심영역 가중치이고, $D_j^{n_i}$ 는 n_i 에서의 가중 MSE(Mean Square Error) 손실이 a^{n_i} , w_b 은 B_j 를 포함하는 서브밴드의 가중치이고, B_j 는 j 번째 코드블록이다. $a(u, v)$ 는 웨이블릿 계수이고, $\hat{a}(u, v)$ 는 n_i 으로 양자화된 웨이블릿 계수이며, n_i 는 절단점을 의미한다. 다음은 코드블록의 우선적 처리 알고리듬을 나타낸 것이다.

3.4 우선적 처리 결과

그림 7은 그림 7의 (a)와 같이 관심영역 마스크가 정의가 되었을 때 기존 방법과 제안한 방법의 우선적 처리 결과를 나타낸다.

점선은 사용자에 의해 지정된 관심영역 모양을 의미하고, 흰색 부분은 배경으로 처리되는 부분을 의미

한다. 진회색 부분은 관심영역으로 우선적 처리되는 부분을 의미하고, 연회색 부분은 배경 웨이블릿 계수

```

01 : static mode
  {
02 : generate ROI mask in the wavelet domain
03 : identify ROI code blocks using ROI threshold
    value
  }
04 : dynamic mode
  {
05 : code LL 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 using ROI threshold
    value
  }
09 : for each code block
  {
10 : for bitplane = MSB : LSB
  {
11 : multipass coding {
12 : encode each pass
13 : calculate coding pass rate R
14 : calculate coding pass distortion estimate D
15 : if code block ∈ ROI code block
16 :   D = D * Wroi
      } //end
    } //end bitplane
17 : } //end code block
18 : rate control to pack bit stream into
    quality-layered structure
  
```

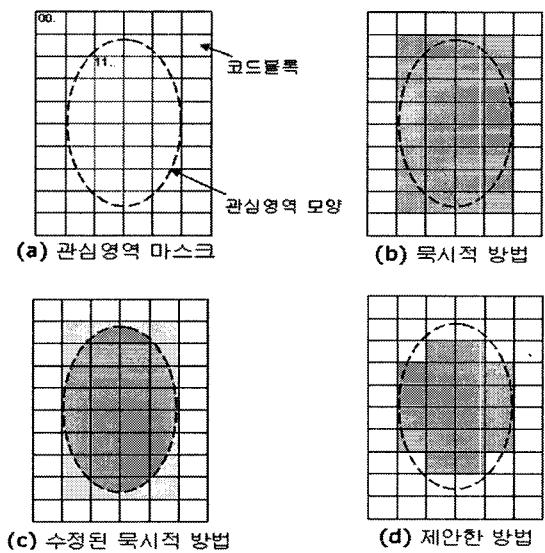


그림 7. 우선적 처리 결과

의 우선권에 의해 조절된 부분을 의미한다. (b)의 목시적 방법은 관심영역 주위의 많은 배경 부분이 우선적으로 처리되는 것을 확인할 수 있고, (c)의 수정된 목시적 방법은 배경 부분에 손실을 주어 우선권에 의해 조절되는 것을 확인할 수 있다. (d)의 제안한 방법은 각 코드블록의 관심영역 포함율이 관심영역 임계값보다 큰 경우에만 우선적 처리가 되는 것을 확인할 수 있다.

4. 실험 및 평가

실험에 사용된 코딩 방법은 관심영역 코딩을 하지 않은 EBCOT, Maxshift 방법, 목시적 방법, 수정된 목시적 방법, 제안한 방법이다. 동일한 실험 조건을 위해 목시적 방법, 수정된 목시적 방법, 제안한 방법에서는 W_{ROI} 의 값을 4096로 설정한다. 수정된 목시적 방법에서는 k 의 값을 5로 설정한다. 제안한 방법에서는 관심영역 코드블록 임계값을 90으로 설정한다. 관심영역 모양은 모두 사각형이며, 관심영역 위치는 이미지의 중앙 부분이다. 관심영역 크기는 전체 이미지의 20%이다. 실험 이미지는 표 1과 같으며, 모두 그레이 이미지(8 bpp)이다.

4.1 주관적 화질 평가

그림 8은 0.125bpp에서 다양한 관심영역 코딩 방법을 사용하여 재구성된 이미지들이다. (a)는 원 이미지를 나타내고, (b)는 관심영역 코딩을 하지 않은 EBCOT 방법으로서 전체 이미지가 희미하게 보이며, (c)는 Maxshift 방법으로서 관심영역 부분만 고품질로 보인다. (d)와 (e)는 수정된 목시적 방법과 제안한 방법으로서 고품질의 관심영역과 저품질의 배경을 보이고 있으며, 두 방법은 눈으로 봤을 때 큰 차이가 없다. 그림 9는 제안한 방법을 이용하여 0.125, 0.5, 1.0, 2.0bpp로 재구성된 Boat 이미지이다. 저비트율에서 전체 이미지에 대한 압축 성능이 뛰어났다.

표 1. 실험 이미지

파일명	해상도	종류
Boat	512 × 512	자연 이미지
Barbara	512 × 512	자연 이미지

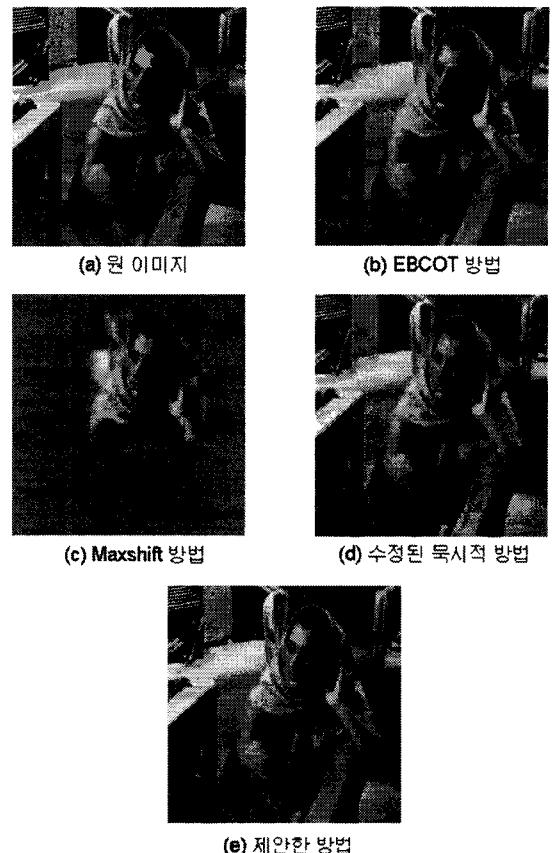


그림 8. 0.125bpp에서 다양한 방법으로 재구성된 “barbara” 이미지

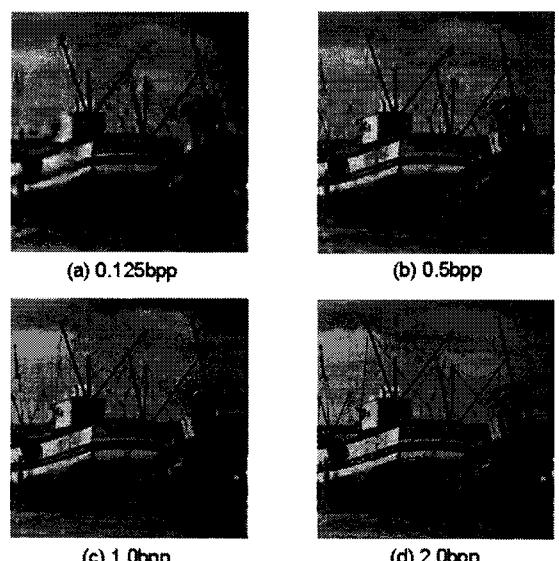


그림 9. 제안된 방법으로 재구성된 “Boat” 이미지

4.2 객관적 화질 평가

본 논문에서는 PSNR를 사용하여 객관적 화질 평가를 하였다. 샘플당 n 비트의 길이를 가지는 이미지에 대한 PSNR은 식(3)으로 나타낸다.

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

MSE를 구하는 함수는 식(4)와 같다.

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

여기서 x 는 원 화소 값을 의미하고, \hat{x} 은 재구성된 화소 값을 의미하며, A 은 관심영역 영역을 의미한다. 관심영역 PSNR은 관심영역의 비트율-왜곡을 측정하여 구할 수 있다. 표 2는 Barbara 이미지에 대한 각 비트율에서 다양한 코딩 방법에 의한 관심영역의 PSNR 값을 나타내고 있다.

표 2와 같이 Maxshift 방법은 관심영역을 우선적 처리를 한 후에 배경을 처리하기 때문에 가장 우수한 값을 나타냈다. 반면에 제안한 방법은 목시적, 수정된 목시적 방법에 비해 약 3% 더 높은 PSNR 값을 나타냈다. 표 3은 무손실 관심영역의 재구성을 위해 요구되는 비트율을 나타낸다.

표 2. 다양한 방법에 대한 관심영역 PSNR 비교

코딩방법 비트율	EBCOT	목시적	Maxshift	수정된 목시적	제안한 방법
0.0625	20.413	21.200	24.885	22.668	23.364
0.125	22.056	24.299	27.289	24.943	25.821
0.25	24.797	27.919	30.876	29.717	30.467
0.5	28.620	32.897	36.014	34.288	35.617
1.0	34.871	40.230	44.733	42.642	43.987

표 3. 무손실 관심영역의 재구성을 위해 요구되는 비트율

관심영역 코딩 방법	비트율	
	20% 관심영역	8% 관심영역
EBCOT	2.242	2.242
목시적	1.443	0.704
Maxshift	1.087	0.326
수정된 목시적	1.127	0.486
제안한 방법	2.242	2.242

관심영역의 크기가 전체 이미지의 20%인 경우 관심영역의 완전한 재구성은 EBCOT는 2.242bpp, 목시적 방법은 1.443bpp, 수정된 목시적 방법은 1.127bpp로 이루어지며, 제안한 방법은 일부 관심영역을 배경으로 처리하기 때문에 EBCOT 방법과 동일한 비트율을 나타낸다. 하지만 제안한 방법은 목시적 방법과 마찬가지로 JPEG2000 표준 관심영역의 무손실 코딩과는 호환이 가능하다. 표 4와 같이 관심영역의 크기가 작을수록 관심영역의 무손실 재구성 비트율도 낮아진다.

4.3 관심영역의 코딩 시간 비교

동적 관심영역은 디코딩 과정에서 관심영역 코딩을 하여 복원을 하기 때문에 빠른 복원을 위해서는 관심영역 코딩 속도가 매우 중요하다. 관심영역 코딩 시간은 사용자의 관심영역 모양 정의 시간과 관심영역 코드블록 판별시간 및 우선적인 처리시간의 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 반면, 제안한 방법은 그림 5와 같이 일부 웨이블릿 계수만 스캔한다. 표 4는 관심영역 코드블록 판별을 위해 코드블록당 평균 스캔 횟수를 비교한 것이다.

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 마스크를 스캔한다. 반면에 제안한 방법은 4번의 스캔과정을 통해 코드블록 판별을 할 수 있었다.

배경 웨이블릿 계수를 포함하는 관심영역 코드블록 판별을 위해 목시적 방법은 계산 편리를 위해 전체 스캔 수의 반으로 하고, 수정된 목시적 방법에서는 관심영역 코드블록 내의 배경 웨이블릿 계수에 대한 우선권 조절을 위해 모든 코드블록의 마스크를 스캔한다. 제안한 방법은 평균적으로 $4 + (n-6)$ 번의 스캔과정을 통해 알 수 있었다. 만약 관심영역 크기

가 전체 이미지의 25%인 경우, 배경 코드블록은 약 70%, 관심영역 코드블록은 약 20%, 배경 웨이블릿 계수를 포함한 관심영역 코드블록은 약 10% 정도 차지하게 되며, 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 약 88%, 관심영역 코드블록은 약 5%, 배경 웨이블릿 계수를 포함한 관심영역 코드블록은 약 7% 정도 차지하게 되며, 코드블록의 가로 크기가 n 이라면, 3 가지 방법의 코드블록당 평균 스캔 수는 $0.915n^2 + 0.05$, n^2 , $0.07n + 3.58$ 이 된다. 결론적으로 제한한 방법이 기존의 방법들과 비교하여 판별을 위한 스캔 속도가 가장 빠르다는 것을 알 수 있었다.

4.4 관심영역 코딩 방법 비교

표 5는 관심영역 코딩 방법의 성능에 영향을 미치는 주요 매개변수에 따른 관심영역 코딩 방법을 비교한 것이다. ①은 정적 관심영역으로서 웨이블릿 계수 단위로 우선적인 처리를 하고, 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 ①, ②, ④는 가능하며, ③은 관심영역 코드블록 내의 모든 배경 손실 처리로 인해 불가능하다. 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, ③의 순으로 빠르게 된다.

표 4. 코드블록당 평균 스캔 횟수 비교

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

n : 코드블록의 가로크기

표 5. 관심영역 코딩 방법 비교

코딩방법 매개변수	①Maxshift	②목시적	③수정된 목시적	④제안한 방법	
정적	동적				
우선적 처리단위	웨이블릿 계수	코드 블록			
관심영역 정확도	○	△	○	△	
무손실로 재구성	○	○	×	○	
관심영역 중요도조절	×	○	○	○	
관심영역 코딩속도	의미없음	□	△	○	

○: 고 □: 중 △: 저 :지원안함

5. 결론 및 향후과제

본 논문에서는 압축된 비트 스트림으로부터 사용자 위주의 관심영역을 추출하여 빠르게 관심영역을 코딩한 후에 이미지를 전송하여 복원하는 동적 관심영역 코딩 방법을 취급하였다. 기존 동적 관심영역 코딩 방법은 관심영역 코딩을 위해 많은 스캔 과정을 필요로 하고, 관심영역 주위에 있는 배경을 우선적으로 처리하는 문제와 손실 처리 문제로 인해 빠른 코딩이 이루어지지 못하였다. 이러한 문제점들을 개선하기 위하여 본 논문에서는 관심영역 모양 특징을 기반으로 효율적인 관심영역 코드블록 판별 알고리듬을 제안하였다. 이 알고리듬은 관심영역 임계값에 따라 성능을 조절할 수 있었다. 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 비교하여 성능 면에 있어서 우수함을 나타냈다. 향후 과제로서 실제로 사용자가 관심영역을 정의한 이후부터 관심영역 코딩, 전송, 복원까지의 시간을 비교하는 실험과 제안한 방법을 Motion-JPEG에 적용하는 사례 연구가 남아있다.

참 고 문 헌

- [1] M. Rabbani and R. Ioshi, "An Overview of the JPEG2000 Still Image Compression Stand-

- ard," *Signal Processing : Image Communication*, Vol.17, pp. 3-48, 2002.
- [2] Athanassios N. Skodras and Touradj Ebrahimi, "JPEG2000 image coding system theory and applications," *IEEE International Symposium on Circuits and Systems*, pp. 3866-3869, May 2006.
- [3] 강기준, 이부권, 서영건, "JPEG2000 용용에서 적절한 ROI 코딩 방법을 적용하기 위한 매개변수의 성능 평가," *정보처리학회 논문지B*, 제 13-B권 제3호, 2006년 6월.
- [4] H-S Kong, A. Vetro, T. Hata, and N. Kuwahara, "Fast Region-of-Interest Transcoding for JPEG2000 Images," *Mitsubishi Electric Research Laboratories, Inc.*, Dec. 2005.
- [5] Andrew P. Bradley and Fred W. M. Stentiford, "JPEG 2000 and Region of Interest Coding," *Digital Image Computing Techniques and Applications*, Jan. 2002.
- [6] 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*, Dec. 2000.
- [7] 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.
- [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] 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.
- [11] Rene Rosenbaum and Heidrun Schumanm, "Flexible, Dynamic and Compliant Region of Interest coding in JPEG2000," *IEEE ICIP*, pp. 22-25, Sep. 2002.
- [12] 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.

강 기 준



2000년 경상대학교 컴퓨터과학
과 학사
2002년 경상대학교 교육대학원
컴퓨터교육과 석사
2007년 경상대학교 대학원 컴퓨터과학과 박사
2006년~현재 거상테크놀로지
근무

관심분야 : JPEG2000, MPEG, 원격교육

안 병 태



1999년 2월 국민대학교 컴퓨터
과학부(이학사)
2001년 2월 경남대학교 컴퓨터
공학부(공학석사)
2006년 8월 경상대학교 컴퓨터
과학부(공학박사)
2006년~현재 유한대학 경영정보
학과 교수

관심분야 : 데이터베이스, 유비쿼터스, XML