韓國컴퓨터情報學會 論文誌
第15卷 第6號, 2010. 6.

2010-15-6-2-1

JPEG2000에서 마스크 패턴을 이용한 빠른 관심영역 처리 기법

이 점 숙*, 하 석 운**, 박 재 흥*, 서 영 건***, 강 기 준*, 홍 석 원*, 김 상 복*

A Rapid Region-of-Interest Processing Technique using Mask Patterns for JPEG2000

Jum Sook Lee*, Seok Woon Ha**, Jae Heung Park*, Yeong Geon Seo***, Ki Jun Kang*,
Seok Won Hong*, Sang Bok Kim*

요 약

JPEG2000 이미지에서 사용자의 관심영역에 따라 동적으로 이미지의 일부를 우선적으로 처리하는 것이 관심영역 처리기법이다. 작은 이미지는 큰 의미가 없지만, 큰 이미지에서는 화면에 출력되는 속도가 느리기 때문에 사용자가 먼저 보고자하는 영역을 지정함으로써 지정된 부분을 우선처리하게 된다. 사용자는 대략의 이미지 중에서 관심영역을 지정하면 지정된 영역의 범위를 마스킹 하여 이미지를 전송한 곳으로 보내게 된다. 관심영역 마스킹 정보를 얻은 서버는 우선적으로 마스킹 되어 있는 코드 블록을 우선적으로 전송한다. 여기서, 빠르게 마스킹 정보를 생성하는 것이 중요한데, 본 연구에서는 미리 만들어 놓은 48개의 마스킹 패턴을 사용하여, ROI(Region-of-Interest)와 배경의 분포에 따라 마스킹 패턴 중에 하나를 선택함으로써 마스킹 영역을 계산하는 시간을 현저히 줄였다. 이 패턴이 적용되는 블록은 한 블록 내에 ROI 영역과 배경 영역이 섞여 있는 블록이다. 한 블록 전체가 ROI 이거나 배경이면 이 패턴은 사용되지 않는다. 실험한 결과, ROI와 배경을 정확하게 분리하여 처리하는 방법에 비하여 약간의 품질은 떨어지지만, 처리시간은 현저히 줄었음을 보였다.

Abstract

An region of interest processing technique is to handle preferentially some part of an image dynamically according to region of interest of the users in JPEG2000 image. A small image is not important, but in a big image the specified region that the user indicated has to be handled preferentially because it takes long time to display the whole image. If the user indicates a region of the outline image, the browser masks the region and sends the mask information to the source that transmitted the image. The server which got the mask information preferentially sends the code blocks matching the masks. Here, quickly generating mask information is important, so, in

• 제1저자 : 이점숙
• 투고일 : 2010. 04. 05, 심사일 : 2010. 04. 06, 게재확정일 : 2010. 04. 29.
* 경상대학교 컴퓨터과학과, 컴・정보통신연구원 ** 교신저자, 경상대학교 정보과학과, 컴・정보통신연구원
*** 경상대학교 컴퓨터교육과, 컴・정보통신연구원
※ 본 연구는 2008년도 경상대학교 광대역연구개발망 활성화를 위한 연구과제 지원비를 받아 수행되었음.

this paper using predefined 48 mask patterns, selecting one of the patterns according to the distribution of ROI(Region-of-Interest) and background, we remarkably reduced the time computing the mask region. Blocks that the patterns are applied are the blocks mixed of ROI and background in a block. If a whole block is an ROI or a background, these patterns are not applied. As results, comparing to the method that precisely handles ROI and background, the quality is unsatisfactory but the processing time remarkably reduced.

▸ Keyword : JPEG2000, ROI, ROI Mask, ROI pattern

Ⅰ. 서 론

현재 컴퓨터에서 이용되는 원시 데이터 중에서 가장 많은 양을 차지하는 것은 영상 데이터이다. 영상은 다양한 응용 분야에서 이용되고 있지만[1], 데이터 량이 많다는 특성으로 인해 제한점을 많이 갖고 있다. 이 제한점을 극복하려면 데이터 량을 줄이는 것과, 먼 곳으로 오류 없이 빠르게 전송하는 것이다. 이런 분야는 이미 많은 연구가 되어 왔고[2,3], 최근에는 많은 데이터 량 중에서 특정 부분만 미리 빠르게 볼 수 있게 하는 관심 영역(ROI : Region of Interest) 부호화가 생겨나게 되었다.

JPEG2000에서는 새로운 정지영상 압축 표준을 발표하여[4,5], 다양한 사용자의 요구 사항을 충족시킬 수 있게 되었다. JPEG2000의 특징은 무손실/손실 압축, 무손실 코딩에의 내포된 손실, 화소 정확성과 해상도에 의한 점진적인 전송, 비트 에러 그리고 관심영역 부호화 등이 있다[6,7]. 특히 관심영역 부호화는 영상 내의 관심 영역을 배경보다 먼저 전송하여 사용자가 볼 수 있게 할 수 있으며, 낮은 압축률로 배경보다 고품질로 영상을 저장하는 것을 말한다. 이렇게 하면, 사용자 입장에서 원하는 부분을 먼저 볼 수 있으며, 그리고 고품질의 영상으로 볼 수 있다. 또한 저비트율의 통신 환경에서는 전체 영상을 모두 받지 않고 원하는 관심영역만 볼 수도 있게 한다[8-10].

언제 ROI를 지정하느냐에 따라서 ROI 부호화 방법은 정적 ROI와 동적 ROI로 나뉜다. 정적 ROI 방법은 이미지 인코딩 과정에서 ROI 마스크를 생성한 후 웨이블릿 계수 단위로 우선적 처리하는 방법이다. 이 방법은 인코딩 시에 ROI가 이미 결정되어 압축되는데, 일반적으로 이미지 전송은 압축 후에 필요에 의해 수행되기 때문에 ROI 코딩 시간이 이미지 전송 시간에 직접적으로 영향을 미치지는 않는다. 표준에서는 Maxshift[9]와 Scaling based[10] 방법을 제안하고 있다.

동적 ROI 방법은 인코딩된 비트스트림의 일부를 사용자에게 전송한 후, 사용자가 관심영역을 지정하면, 그 때 ROI가 지정된다. 이 방법에는 표준에서 제안하고 있는 묵시적(Implicit)[9] 방법이 있으며, 동적 ROI 기법에서는 무엇보다도 ROI 마스크를 빠르게 생생해야 한다. [11]에서는 하나의 블록 내에서 ROI와 배경 영역을 구분하는데, 가장자리를 탐색하여 배경과 ROI가 나뉘는 두 점을 얻어, 두 점을 직선으로 연결하여 개략적인 마스크를 생성하였다. 이 방법은 약간의 정밀성을 떨어뜨리면서 마스크 생성 속도를 높이고자 하였다. 결국 이런 방법들은 사람이 이미지를 보고 관심 영역을 결정하는데 비해, 자동적으로 관심 영역을 추출하는 연구도 활발히 진행되고 있다[12].

본 연구에서는 ROI 마스크를 빠르게 생성하여 서버로 전송하는 기법을 제안하고, 실험하였다. ROI 마스크는 한 블록의 전체가 ROI인 경우는 ROI 마스크 처리되며, 한 블록 내에 ROI와 배경이 섞여 있는 경우에는 일부가 ROI 마스크로 처리 되어야 한다. 이렇게 두 가지가 섞여 있는 경우에 빠르게 ROI 마스크를 계산하고 서버로 전송하는 데, 화질은 최대한 보장해 주는 것이 중요하다. 이를 위해 본 연구에서는 한 블록에서 가장자리 8개의 점을 이용하여 개략적인 ROI 마스크를 생성한다. 다른 ROI 마스크 생성 기법들은 ROI 마스크를 생성하기 위해 한 블록을 모두 탐색하면서 마스크 테이블은 만들지만, 제안한 기법은 미리 정의된 48 개의 마스크 패턴을 사용함으로써 마스크 테이블은 만드는 시간을 혁신적으로 줄였고, ROI와 배경의 경계점을 찾는데 8개의 점만을 사용함으로써 경계선을 찾는 계산 시간도 혁신적으로 줄였다. 다만 그 경계점이 대략의 경계를 이루고 있으므로 화질은 다른 기법들이 비해 다소 떨어진다.

Ⅱ. 관련연구

ROI 부호화는 전체 영상을 전송 및 복원하기보다는 ROI를 배경보다 우선적 처리하는 기술이다. 이것은 점진적 이미지 전송 방법보다 압축률과 전송시간을 단축시킬 수 있을 뿐만 아니라 효율적인 메모리 관리 등으로 인해 보다 빠른 이미

지 서비스가 가능하다[13, 14]. ROI 마스크 생성은 사용자가 원 이미지를 보고 ROI 모양을 정의하면, 이미지 도메인에서의 이진 ROI 마스크를 생성하고, 생성된 ROI 마스크는 IDWT(Inverse DWT)를 이용하여 웨이블릿 도메인에서의 ROI 마스크를 생성한다. 이렇게 구한 ROI 마스크 정보에 의해 ROI 웨이블릿 계수 단위로 중요도에 따라 업 스케일링함으로서 우선적 처리를 한다.

대부분의 응용 분야에서는 ROI 서비스를 위해 정적 ROI 코딩 방법으로도 충분히 가능하다. 하지만 인코딩 과정에서 ROI를 모른다면, 디코딩 과정에서 ROI 모양 정보를 받아서 ROI 코딩을 한다. 이 처럼 디코더 과정에서 ROI 모양을 정의한 후 ROI 코딩하는 방법을 동적 ROI 코딩이라고 한다[8, 9]. 동적 ROI 코딩 과정은 ROI 코딩이 되어 있지 않은 압축된 비트 스트림으로부터 기본적인 압축 정보와 LL 밴드 내용을 추출하여 디코더로 전송한다. 사용자는 복원된 LL 밴드 내용을 보고 ROI를 정의하고 이 정보를 인코더로 전송한다. 인코더에서는 ROI 코드블록 혹은 ROI 패킷을 판별하여 최종 ROI 마스크를 생성한 다음에 ROI 코드블록이나 패킷을 ROI 중요도에 따라 우선순위를 할당하여 ROI 코딩을 한다.

본 연구와의 비교를 위하여 Maxishift 방법, 묵시적 ROI 방법, 수정된 묵시적 ROI 방법, 기울기 정보기반 방법[11]을 사용한다.

1. Maxshift ROI 코딩

Maxshift방법은 양자화된 계수 중에 ROI 계수와 배경 계수를 구분하여, 배경 계수 중에서 가장 큰 계수 값, s을 구한 다음, ROI 계수를 s보다 높은 비트-평면에 이동시키는 방법으로서 JPEG2000 Part1 표준이다. 식(1)은 s을 구하는 수식이고, 식(2)는 ROI 처리 후의 계수, $a'(u,v)$를 구하는 수식이다.

$$s \geq \max(M_b) \quad \cdots\cdots (1)$$

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

식(1)에서 $\max(M_b)$은 각 서브밴드에서 양자화된 배경 계수 중에 가장 큰 값을 의미한다. 식(2)에서 $M(u,v)$는 ROI 마스크 정보로서 계수가 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.

2. 묵시적 ROI 코딩

EBCOT에서 각 품질 레이어는 코드블록들의 임베디드 비트 스트림으로부터 임의의 공헌도를 포함한다. 따라서 우선처리는 각 코딩 패스에서 손실률을 조절한 후, PCRD 최적화 알고리즘을 다시 수행한다. 전체 손실 최소화에 의해서 코드블록 공헌도를 할당하기 때문에, 묵시적 ROI는 손실 감소와 ROI가 일치하도록 코드블록 공헌도를 할당한다. 식(3)은 손실 계산방법을 나타낸다.

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

$$\cdots\cdots (3)$$

식(3)에서 W_{ROI}은 가중치이고, $D_j^{n_i}$은 n_i에서의 가중 MSE 손실이고, w_b은 B_j을 포함하는 서브밴드의 가중치이고, B_j은 j번째 코드블록이고, $a(u,v)$은 계수이고, $\hat{a}(u,v)$은 n_i으로 양자화된 계수이고, n_i은 절단점을 의미한다.

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

3. 수정된 묵시적 ROI 코딩

이 방법은 알고리즘 복잡도 없이 ROI 코드블록에 포함된 배경 계수의 우선권을 줄여 묵시적 방법을 보안한 방법이다. ROI 코드블록 내의 배경 계수의 우선권 조절은 배경 계수의 k LSB만큼 절단하 방법으로 실현된다. 식(4)는 ROI 처리 전의 ROI 코드블록내의 계수인 $a(u,v)$을 우선권 조절 후의 계수인 $\bar{a}(u,v)$로 변환하는 식이다.

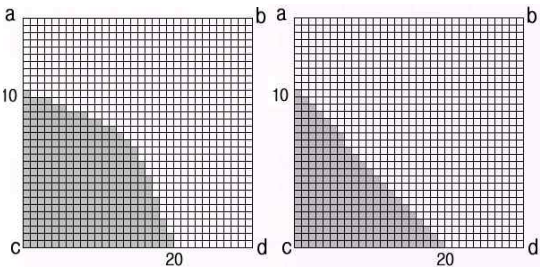
$$\bar{a}(u,v) = \begin{cases} sign[a(u,v)] \left\lfloor \frac{|a(u,v)|}{2^k} \right\rfloor 2^k, & BG \text{ 계수} \\ a(u,v), & ROI \text{ 계수} \end{cases} \quad \cdots\cdots (4)$$

매개변수 k는 인코더에서 지정되며, 조정이 가능하다. k 값이 클수록 ROI 코드블록내의 배경 계수의 우선권이 낮아진

다. 장점은 계수 단위로 ROI 처리가 가능하며, 전체 ROI 전송이 묵시적 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.

4. 기울기정보 기반 방법

한 코드 블록 내의 계수를 전부 탐색하지 않고 일부만 탐색하여 개략적인 정보를 이용해 빠르게 ROI 마스크를 생성하는 기법이다. 개략적인 정보는 코드 블록의 가장자리에 걸쳐 있는 ROI와 배경간의 경계를 빠르게 탐색한 후, 탐색된 두 가장 자리의 위치를 직선으로 연결하여 한쪽은 배경, 다른 한쪽은 ROI 영역으로 간주한다. 그림 1의 (a)의 회색 부분은 ROI 영역으로 지정된 부분이고 오른쪽은 배경이다. (b)는 (a)에서 나누어진 영역을 경계선으로 그은 그림을 보이고 있다. 이 방법은 빠르지만 정밀도는 조금 떨어진다.



(a) 원래 경계 (b) 경계선 구분

그림 1. [11] 방법의 배경과 ROI 구분 예
Fig. 1. An example dividing ROI and background of [11]

본 연구에서 제안하는 방법은 한 블록에서 가장자리 8개의 점을 이용하여 개략적인 ROI 마스크를 생성하는 미리 정의된 48 개의 마스크 패턴을 사용하여 마스크 테이블을 만드는 방법을 제안한다.

III. 가장자리 8 점을 이용한 마스크 생성

JPEG2000에서 ROI처리를 위한 전체 구조는 그림 2와 같으며, ROI 처리를 위하여 디코더에서 실행되는 부분만 본 연구에 해당된다.

1. 관심영역 지정과 ROI 코딩

인코더 또는 서버에서 전송된 이미지는 개략적인 이지인 LL 밴드만 우선 전송되며 사용자는 이 개략적인 이미지를 보고 우선적으로 보고자 하는 영역을 표시 한다. 이렇게 표시된 영역을 관심영역이라 하며, 지정된 관심영역에 관한 정보는 ROI 정보이며 이 정보가 서버로 전송되어 ROI 코딩된다. ROI 코딩된 이미지는 우선적으로 사용자에게 전송되어 보여진다. 이 과정에서 빠른 처리를 요하기 때문에 정확한 ROI 영역을 표시하기보다 약간의 품질은 떨어지더라도 빠르게 처리하는 것이 중요하다[11].

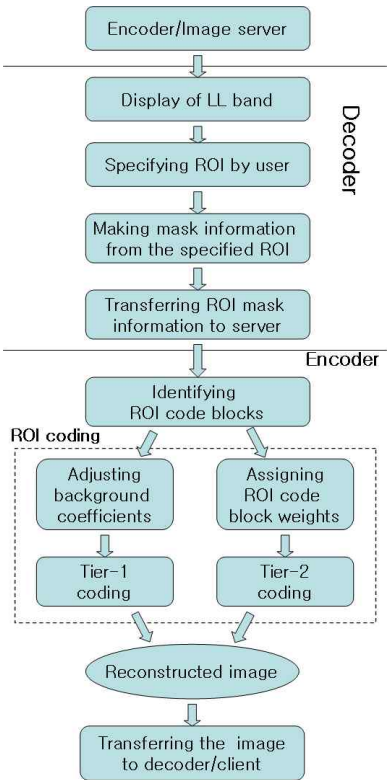


그림 2. ROI 코딩을 지원하는 JPEG2000
Fig. 2. JPEG2000 supporting ROI coding

화면에 표시된 이미지는 웨이블릿 도메인에서의 위치와는 다르기 때문에, 이를 웨이블릿에 적용하기 위해서는 웨이블릿 도메인에서의 마스크 정보로 변환해야 한다. ROI에 포함되는 마스크 정보는 식(5)와 같이 구성된다. 값이 1이면 ROI에 해당되는 픽셀을 의미하고 0이면 배경이 된다.

$$M(x,y)=\begin{cases}1, & \text{관심영역}\\ 0, & \text{배경}\end{cases} \quad \cdots\cdots (5)$$

웨이블릿 도메인에서의 ROI 마스크 정보는 이미지 도메인에서의 ROI 마스크 정보와 달라서, IDWT를 이용하면 구할 수 있다. 즉, 웨이블릿 도메인에서의 ROI 마스크 생성 과정은 우선 이미지 도메인에서의 ROI 마스크와 마지막 IDWT에 의해 2개의 서브밴드 안에 어떤 위치의 계수가 ROI 계수로 재구성하기 위해 사용되었는지를 추적하게 된다. 그런 후에, 두 서브밴드에서의 ROI 마스크와 마지막 이전의 IDWT에 의해 각각 2개의 서브밴드 안에 어떤 위치의 계수가 ROI 계수로 재구성하기 위해 사용되었는지를 추적하게 된다. 이렇게 하여 각 분해레벨에서 모든 서브밴드 내에 어떤 위치의 계수가 ROI 계수로 재구성하기 위해 사용되었는지를 추적하게 된다.

2. ROI 마스크 패턴

본 연구에서 사용되는 코드블록은 32x32로 사용하지만, 64x64를 해도 상관이 없다. 다만 코드블록이 커지면 ROI의 정밀도가 떨어진다. ROI 영역을 표시하기 위하여 ROI 마스크를 정의하는 데, 정확한 영역을 정의하려면 묵시적 방법[9]을 써야 한다. 한 코드블록이 모든 픽셀을 탐색하여 정확하게 ROI를 정의한다는 것은 많은 시간이 소요될 뿐 아니라, 사람의 눈으로는 정확한 것과 유사한 것을 거의 구분하지 못한다. 그러므로 오히려 ROI 영역을 개략적으로 정의하고 빠른 시간 안에 마스크를 생성하는 것이 중요하다. 본 연구에서는 32x32 ROI 마스크를 미리 정의하여 개략적인 ROI 마스크를 생성한다. 사용되는 마스크 패턴은 48 개이며 표 1과 같다.

표 1에서 abcd는 각 코드블록의 꼭지점을 의미하고 efgh는 가장자리 변의 중간점을 의미한다. 각 점의 위치는 그림 3과 같다. 각 값이 갖는 값 중 1은 ROI로 표시된 것이며 0은 배경이다. x는 efgh 변의 값 중에서 정해지지 않은 두 값을 의미한다. 정해진 값 두 개의 값은 1 또는 0으로 표시되며, 두 x의 값은 00, 01, 10, 11의 값을 가지며 세부 패턴에 해당된다.

표 1. ROI 분포에 따른 탐색 순서
Table 1. Searching Order with ROI distribution

a b c d	대표 패턴	e f g h	세부 패턴 (00 01 10 11)
0 0 0 0		0 0 0 0	배경
1 1 1 1		1 1 1 1	ROI
0 0 0 1		0 0 x x	
0 0 1 0		0 x 0 x	
0 0 1 1		0 x x 1	
0 1 0 0		x 0 x 1	
0 1 0 1		x 0 1 x	
0 1 1 0		제외	
0 1 1 1		x x 1 1	
1 0 0 0		x x 0 0	
1 0 0 1		제외	
1 0 1 0		x 1 0 x	
1 0 1 1		x 1 x 1	
1 1 0 0		1 x x 0	
1 1 0 1		1 x 1 x	
1 1 1 0		1 1 x x	

표 1에 보인 패턴은 48개이며, 패턴의 검은 부분은 ROI영역을 흰 부분은 배경을 표시한다. 한 블록이 전체 ROI인 경우는 abcd가 1111이 되며, 배경인 경우는 abcd가 0000이 되며, efgh를 평가할 필요가 없다. 또한 제외로 표기된 블록은 양쪽에 ROI가 나뉘어 지는 경우인데 이런 경우는 실제로는 존재하지 않는 것으로 간주한다.

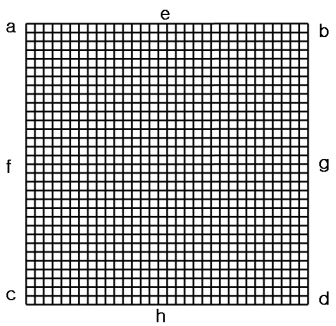


그림 3. 코드블록 판별을 위한 점의 위치
Fig. 3. Positions to get sampling

3. 패턴 찾기

그림 4의 (a)는 한 블록 내에 ROI와 배경이 혼합된 블록을 보이며, 회색 부분은 ROI이며 흰색 부분은 배경이다. (a)의 블록에 대해서 패턴 찾기를 하면 abcd = 0010, efgh = 0101이 되어, (b) 패턴이 찾아 진다. (b)의 검은 부분은 ROI로 코딩되고, 흰색부분은 배경으로 코딩된다. 다음은 패턴을 찾는 알고리즘이다.

```
get a, b, c and d from mask bit table
id = a<<3 or b<<2 or c<<1 or d ;
if (id is 0) current_pattern = 0 ;
else if (id is 15) current_pattern = 49 ;
else begin
     get e, f, g and h from mask bit table ;
     id = a<<7 or b<<6 or c<<5 or d << 4 or
          e<<3 or f<<2 or g<<1 or h ;
   current_pattern = conversion_tab(id) ;
end
```

위 알고리즘에서 “or”는 비트 단위의 논리합을 의미하고, ”<<“는 왼쪽으로 숫자만큼 비트 단위로 옮기는 것을 의미한다. ”mask bit table“은 사용자가 그림 4의 (a)처럼 선택을 했다면 회색부분은 1로 채워지고 흰색부분은 0으로 채워진 한 코드 블록의 비트 테이블을 의미한다. 패턴은 모두 48 개로 구성되며 1에서 48까지의 인덱스가 곧 패턴인덱스이다. 0은 모두 배경인 패턴 번호를 의미하고 49는 모두 ROI로 구성된 코드 블록의 패턴 인덱스를 의미한다. 0과 49는 e,f,g,h를 검사하지 않고 패턴이 결정된다. id는 8비트로 구성되며 각 비트는 a,b,c,d,e,f,g,h의 이진 값을 가지므로 48 개의 값들은 표 1로부터 표 2에 나타낸 값을 가지게 된다. 이 값들은 패턴 번호와 일치시키기 위하여 conversion_tab() 함수를 사용한다. 예를 들어, a=1, b=0, c=1, d=0 이면, f=1, g=0이 된다. 여기서 e=0, h=1 이면 id = 10101001(2) 이 되고, 세부패턴 xx(eh)=01 이므로 id 값은 165가 된다. 165는 conversion_tab() 함수에 의해 31번 패턴 번호가 선택된다. 이렇게 얻어진 패턴 번호는 인코더로 전송되어 ROI 코딩에 사용된다.

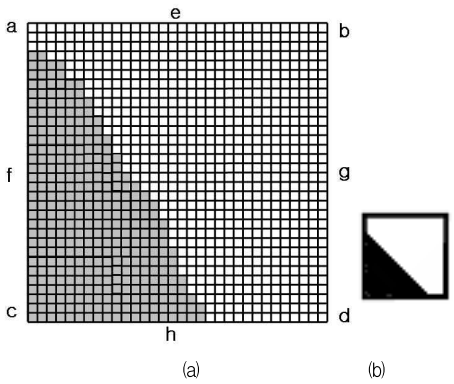


그림 4. 샘플 코드 블록(a)와 패턴(b)
Fig. 4. Sample code block(a) and its pattern(b)

표 2. id 값으로부터 패턴 번호 얻기
Table 2. Getting a pattern number from id

id (8비트)		세부 패턴(xx) (00 01 10 11)	conversion _tab()
a b c d	e f g h		
0 0 0 1	0 0 x x	16 17 18 19	1 2 3 4
0 0 1 0	0 x 0 x	32 33 36 37	5 6 7 8
0 0 1 1	0 x x 1	49 51 53 55	9 10 11 12
0 1 0 0	x 0 x 1	65 67 71 75	13 14 15 16
0 1 0 1	x 0 1 x	82 83 90 91	17 18 19 20
0 1 1 1	x x 1 1	115 119 123 127	21 22 23 24
1 0 0 0	x x 0 0	128 132 136 140	25 26 27 28
1 0 1 0	x 1 0 x	164 165 172 173	29 30 31 32
1 0 1 1	x 1 x 1	181 183 189 191	33 34 35 36
1 1 0 0	1 x x 0	200 202 204 206	37 38 39 40
1 1 0 1	1 x 1 x	218 219 222 223	41 42 43 44
1 1 1 0	1 1 x x	236 237 238 239	45 46 47 48

4. ROI 마스크 생성과 우선 처리

사용자가 ROI를 지정한 것을 디코더에서는 한 코드블록당 1 개의 마스크 패턴을 찾아서 인코더로 전송하게 된다. 인코더에서는 최종 ROI 마스크 테이블을 만들어 EBCOT과정을 통하여 ROI 영역은 우선 처리되어 디코더로 전송한다. 이 과정은 신속히 처리되어야 하며, 불필요한 계수는 탐색할 필요가 없게 만들어야 한다. 여기서는 이미 48 개의 패턴을 가

지고 있으므로 기울기 기반 방법[11]에서처럼 마스크를 구하기 위하여 추가적인 계산을 할 필요가 없다. 단지 디코더에서 전송된 1 바이트의 마스크 패턴 번호를 이용하여 이미 정해진 마스크를 설정한다.

Ⅳ. 실험 및 평가

실험은 Maxshift 방법, 묵시적 방법, 수정된 묵시적 방법, 기울기정보 기반 방법, 제안한 방법을 이용하여, PSNR과 ROI 마스크 생성 시간들을 비교하였다. 실험을 위해 Maxshift 방법을 제외한 방법에서 k의 값은 5로 설정하였다. 실험에 사용된 영상은 그림 5와 같이 Yuna와 Persimmon이다. 관심영역의 모양은 타원이며, 위치는 이미지의 중앙 부분으로 하였다. 관심영역의 크기는 각각 전체 이미지의 25%와 30%이다.

그림 5. 실험에 사용된 이미지
Fig. 5. Images used for experiments

1. 화질 평가

표준에서 제공되는 부호화 방법들은 마스크 생성을 위해 모두 순차 탐색을 하며, 기울기 기반 방법은 일부 마스크 정보만 탐색한다. 그러나 본 기법은 많아야 6 개의 점을 탐색하고 ROI 마스크도 이미 정해 둔 패턴을 이용하기 때문에 탐색 시간이 현저히 줄어든다. 다만 표 3과 4에 보인 바와 같이 제안한 방법은 화질 면에서는 다른 방법에 비하여 떨어질 수밖에 없다. 그 이유는 빠르게 ROI 코딩을 하지 위하여 근삿값을 사용하였기 때문이다. PSNR을 사용하여 객관적 화질 평가를 하였으며, 샘플당 n 비트의 길이를 가지는 이미지를 위한 PSNR은 식(6)과 같고

$$PSNR = 10\log_{10}\frac{(2^n-1)^2}{MSE} \quad \cdots\cdots (6)$$

MSE는 식(7)과 같다.

$$MSE = \frac{\sum(\hat{x}-x)^2}{A} \quad \cdots\cdots (7)$$

여기서 x는 원 화소 값을 의미하고, $\hat{x}$은 재구성된 화소 값을 의미하고, A는 ROI를 의미하고, 따라서 ROI PSNR은 ROI의 R-D 성능을 측정하여 구할 수 있다. 표 3은 Yuna 이미지, 표 4는 Persimmon을 이용하여 화질을 측정한 것이다.

표 3. 다양한 방법의 ROI PSNR 비교(Yuna)
Table 3. Comparison of ROI PSNR in different methods

코딩방법 / 비트율	Maxshift	수정된 묵시적	기울기 기반방법	제안한 방법
0.0625	28.003	24.834	24.606	24.278
0.125	30.459	28.535	28.345	27.980
0.25	34.076	32.777	32.557	32.341
0.5	39.454	38.487	38.350	37.934
1.0	47.300	46.898	46.347	45.965

표 4. 다양한 방법의 ROI PSNR 비교(Persimmon)
Table 4. Comparison of ROI PSNR in different methods

코딩방법 / 비트율	Maxshift	수정된 묵시적	기울기 기반방법	제안한 방법
0.0625	28.034	24.867	24.634	24.289
0.125	30.149	28.609	28.409	28.023
0.25	34.079	32.779	32.590	32.390
0.5	39.400	38.567	38.409	38.002
1.0	47.387	47.001	46.366	45.997

2. ROI 마스크 생성 시간 비교

기울기 기반 방법을 제외한 다른 방법들은 마스크 생성을 위해 순차 탐색을 하는 반면에, 제안한 방법은 표 5과 같이 일부 마스크 정보만 탐색한다. Maxshift 방법과 수정된 묵시적 방법은 모든 픽셀을 탐색하고, 묵시적 방법은 최악의 경우에 모든 픽셀을 탐색하지만, 첫 번 탐색에서 ROI로 판정되면 그 코드 블록이 모두 ROI 블록으로 간주되므로 1회에 탐색이 끝날 수도 있다. 기울기 기반 방법은 코드 블록이 배경이거나 ROI인 경우에는 제안한 방법과 같은 탐색시간을 가지며, 섞인 코드 블록인 경우에는 앞의 세 방법은 n2에 비례하는 탐색시간을 가지며, 기울기 기반 방법은 n에 비례한다. 반면에 제안한 방법은 a,b,c,d 네 점과 e,f,g,h 중 두 점을 탐색하여 얻어지므로 6 회의 탐색 시간만 갖는다.

표 5. 다양한 방법의 평균 탐색 시간
Table 5. Mean scan times (n : 32)

코딩방법 / 탐색횟수	Max shift	묵시적	수정된 묵시적	기울기 기반 방법	제안 방법
배경 코드블록	n^2	n^2	n^2	4	4
ROI 코드블록	n^2	1	n^2	4	4
배경과 ROI가 섞인 코드블록	n^2	$n^2/2$	n^2	4 + n	6

표 6. 다양한 방법의 마스크 생성 시간
Table 6. Mean mask generation times of different methods

코딩방법 / 탐색횟수	Maxshift	묵시적	수정된 묵시적	기울기 기반방법	제안방법
배경 코드블록	0	0	0	0	0
ROI 코드블록	n^2	n^2	n^2	n^2	1
배경과 ROI가 섞인 코드블록	n^2	n^2	n^2	$n^2/2$	1

표 6은 ROI 마스크 생성 시에 소요되는 시간을 보인다. 제안한 방법은 이미 마스크가 생성되어 있으므로 디코더로부터 얻어진 패턴 번호를 인덱싱하여 찾아가면 된다. 반면에 다른 방법들은 수식이나, 마스크 비트를 사용하여 한 번 더 ROI 마스크를 생성하는 과정을 거쳐야 한다. ROI 정확도는 모든 영역을 탐색하여 하나의 계수 범위까지 마스크를 표현해 주는 Maxshift와 수정된 묵시적 방법이 가장 좋다. 제안한 방법은 마스크 비트 탐색시간이나 마스크 생성시간이 다른 기법에 비해 현저히 줄어든 것에 비해 약간의 품질은 떨어질 수 밖에 없다.

V. 결론 및 향후과제

ROI 부호화는 사용자가 관심 갖는 영역을 우선처리하거나 고화질로 압축하는 것을 말한다. 특히 동적 ROI 코딩에서는 빠른 시간 안에 코딩을 하는 것이 중요하므로 ROI 마스크 탐색시간 생성시간을 줄이는 것이 관건이다. 기존의 ROI 부호화에서는 ROI 마스크 생성을 위하여 많은 탐색 과정을 거친다. 이로 인해 많은 시간이 걸리는 문제와 ROI 주위에 있는 배경을 ROI으로 판별하는 문제 때문에 효율적인 부호화가 어렵다. 시간을 줄이기 위한 방법인 기울기 기반 방법은 기존의 방법보다는 시간을 줄였지만 여전히 시간이 꽤 걸린다. 하지만 제안한 방법에서는 단지 6 개의 가장자리 점만을 탐색하여 탐색시간을 줄였고, 탐색 후 얻어진 패턴 번호를 이용하여 ROI 마스크의 생성시간도 현저히 줄였다. 이 방법은 개략적인 ROI 마스크를 생성하므로 화질은 기존 방법에 비해 떨어지지만 실시간 처리에 유용하게 사용될 수 있다. 향후 과제로는 자동 ROI 추출을 이용하여 동적 ROI 코딩에 적용하는 것이다.

참고문헌

[1] N. Kock, "Media Richness or Media Naturalness? The Evolution of Our Biological Communication Apparatus and Its Influence on Our Behavior Toward E-Communication Tools", IEEE Transactions on Profession Communication, Vol. 48, No. 2, Jun. 2005

[2] M. Rabbani and R. Joshi, "An overview of the JPEG2000 still image compression standard", Signal Processing, Vol. 17, pp.3-48, 2002

[3] J. Hara, "An Implementation of JPEG 2000 Interactive Image Communication System", ISCAS 2005, Vol. 6, pp.5922-5925, May 2005

[4] J. In, Shahram Shirani, and Faouzi Kossentini, "On RD Optimized Processive Image Coding Using JPEG", IEEE Transactions on Image Processing, Vol. 8, No. 11, Nov. 1999

[5] Ahn, C.B., Kim, I.Y., Han, S.W., "Medical Image Compression Using JPEG Progressive Coding", Nuclear Science Symposium and Medical Imaging Conference, 1993 IEEE Conference Record, pp.1336-1339, Nov. 1993

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

[7] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 Still Image Coding System : An Overview", IEEE Transactions on Consumer Electronics, Vol. 46, No. 4, pp.1103-1127, Nov. 2000

[8] Kong H-S, Vetro A., Hata T. and Kuwahara N., "Fast Region-of-Interest Transcoding for JPEG2000 Images", Mitsubishi Electric Research Laboratories, Inc., Dec. 2005

[9] M. Boliek and C. Christopoulos, "JPEG 2000 Part

I Final Committee Draft Version 1.0", ISO/IEC JTC 1/SC 29/WG 1 N1646R, March 2000

[10] M. Boliek, E. Majani, J. S. Houchin, J. Kasner and M. L. Carlander, "JPEG 2000 Part II Final Committee Draft", ISO/IEC JTC 1/SC 29/WG 1 N2000, Dec. 2000

[11] 박순화 외 4명, "관심영역 코딩을 위한 기울기 정보기반의 빠른 마스크 생성 기법", 한국컴퓨터정보학회 논문지, 제 14권, 제 1호, 81-89쪽, 2009년. 1월.

[12] 박재홍 외 4명, "JPEG2000에서 ROI의 자동 추출과 우선적 처리", 한국컴퓨터정보학회 논문지, 제 14권, 제 6호, 127-136쪽, 2008년. 11월.

[13] 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, pp.590-596, Oct. 2005

[14] J. Askelof, M. Larsson and C. Christopoulos, "Region of Interest Coding in JPEG2000", Signal Processing : Image Communication 17, pp. 105-111, 2002

저 자 소 개

이 점 숙

1994년 : 경상대학교 컴퓨터과학과 학사
1999년 : 경상대학교 컴퓨터교육과 석사
2003년~현재 : 경상대학교 컴퓨터과학과 박사수료
관심분야 : MPEG, JPEG2000, wavelet

하 석 운

1978년 : 부산대학교 전자공학과 학사
1990년 : 부산대학교 전자공학과 박사
1990년~현재 : 경상대학교 정보과학과 교수
관심분야 : 임베디드 시스템, 영상인식, 영상처리

박 재 흥

1978년 : 충북대 수학교육과
1989년 : 중앙대 대학원 전산과 박사
1983년~현재 : 경상대 컴퓨터과학과 교수
관심분야 : 소프트웨어 공학, 테스팅, 소프트웨어 신뢰성

서 영 건

1987년 : 경상대학교 전산과 학사
1997년 : 숭실대학교 전산과 박사
1989년~1992년 : 삼보컴퓨터
1997년~현재 : 경상대학교 컴퓨터교육과 교수
2001년~현재 : 경상대학교 컴퓨터정보통신연구소원
관심분야 : 멀티미디어통신, 영상인식, 원격교육

강 기 준

2000년 : 경상대학교 컴퓨터과학과 학사
2002년 : 경상대학교 컴퓨터교육과 석사
2007년 : 경상대학교 컴퓨터과학과 박사
관심분야 : JPEG2000, MPEG,

홍 석 원

2001년 : 경상대학교 멀티미디어과 석사
2003년~현재 : 경상대학교 컴퓨터과학과 박사과정
관심분야 : 영상처리, 데이터 압축

김 상 복

1989년 : 중앙대학교 전자공학과 박사
1984년~현재 : 경상대학교 컴퓨터과학과 교수
2001년~현재 : 경상대학교 컴퓨터정보통신연구소원
관심분야 : 멀티미디어통신, 영상인식, 보안