http://dx.doi.org/10.3745/KIPSTB.2011.18B.6.349

JPEG2000에서 마스크 패턴을 이용한 빠른 동적 ROI 코딩

강 정 현†·서 영 건††

요 약

JPEG2000의 ROI 코딩에서, 사용자가 지정한 큰 이미지의 특정 영역은 우선적으로 처리되어야 한다. 왜냐하면, 전체 이미지를 나타내는데 많은 시간이 소모되기 때문이다. 사용자가 개략적인 이미지의 영역을 지정하면, 브라우저는 그 영역을 마스킹하고, 마스킹 정보를 서버로 전송한다. 마스크 정보를 전송받은 서버는 그에 해당하는 코드 블록들을 우선적으로 전송한다. 여기서, 마스크 정보의 빠른 생성이 중요하다. 이를 위해, 본 논문에서는 48개의 미리 정의된 마스크 패턴을 사용한다. 마스크 패턴은 마스킹된 영역을 빠르게 계산하기 위하여 ROI와 배경의 분포의 모양에 따라 미리 만들어져 있다. 기존의 방법인 ROI와 배경의 모양을 정확하게 나누어 처리하는 방법에 비해 품질은 아주 조금 떨어지지만 처리시간은 월등히 줄어졌음을 본 논문에서는 보인다.

키워드 : JPEG2000, ROI, ROI Coding

Fast Dynamic ROI Coding using the Mask Patterns in JPEG2000

Juong-Hyon Kang† · Yeong-Geon Seo††

ABSTRACT

In ROI processing of JPEG2000, a region of large image indicated by the user must be processed preferentially, because it takes a considerable amount of time to display the full image. When the user indicates a region of the outlined image, then the browser masks the region and sends the mask information to the server that transmitted the outlined image. The server that receives the mask information preferentially sends the corresponding code blocks. Here, a quick generation of mask information is important. In this paper, we use 48 predefined mask patterns, which are defined according to the distribution shape of ROI and background to reduce the computing time. As a result, compared to other methods that precisely handles the ROI and background, the processing time of the method is remarkably reduced, but the quality is short of the existing methods just a little bit.

Keywords : JPEG2000, ROI, ROI Coding

1. Introduction

Most of the primitive data being used in Internet is image data. Images are used for diverse applications [1], but they are constrained by their large amount of data. To cope with this constraint, we reduce the amount of data and quickly transmit data without errors. These fields have been studied [2, 3], and lately, ROI (Region-of-Interest) coding has allowed us to view a specific region of a large amount of data. JPEG2000 has announced a new still image compression standard [4], and it requires users to satisfy their different requirements. The typical characteristics of JPEG2000 are lossy and lossless compression, involved lost in lossless coding, progressive transmission by the pixel precision and the resolution, bit errors and ROI [4 ,5]. In particular, ROI coding transmits the object of interest in the image preferentially and permits users to view it. Furthermore, the object of interest can be stored with higher quality than the image's background. In that case, the users can obtain first what they prefer to see with a high quality image. In addition, in communication environments with low bit rates it allows users to view a specific region of interest without receiving the full image [6].

ROI coding method is classified as static and dynamic ROI method. As static ROI method, JPEG2000 standard recommends Maxshift and Scaling based method [7]. Dynamic method specifies ROI after a part of encoded bit

† 준 회 원 : 경상대학교 경영정보학과 박사수료
†† 종신회원 : 경상대학교 컴퓨터과학과 교수(교신저자)
논문접수 : 2011년 9월 20일
수 정 일 : 1차 2011년 10월 10일
심사완료 : 2011년 10월 10일

streams are transmitted to the user. Namely, if the user specifies a part of interest, the ROI is determined. In this method, the Implicit method [7] is recommended by JPEG2000 standards. In dynamic method, ROI mask must be generated as soon as possible. To process the ROI speedy, a study using the patterns has been proposed, but it was not only to use the patterns [8]. Another study [9] proposed that the outlined information was acquired from searching the boundaries between the ROI and the background. Our study proposes a method that generates ROI masks speedy and transfers them to the server. In the case of mixed ROI block, we need to obtain the ROI masks and send them to the server quickly, to make an effort to guarantee high image quality. For this, we use 8 edge points of a block and roughly generate the ROI mask. Existing approaches generate the mask after looking for all pixels of the block, but the proposed method only uses 48 predefined mask patterns. This method can reduce the time used not only to create the mask table, but also to search the boundary line of the ROI and background using only 8 points. However, the image quality is slightly lower than the others.

2. Related Studies

ROI coding is a technique for handling ROI in a preferential manner over background, but not transferring and decoding the entire image. This can not only reduce the transmission time and the compression rate more than progressive image transmission method, but also can serve them more rapidly due to efficient memory management [10, 11]. Generating an ROI mask occurs after the user specifies an ROI shape using the original image. The binary ROI masks are generated in the image domain, and the ROI masks are completed in the wavelet domain using IDWT. Then, the preferred processing is handled by up-scaling the coefficients by the ROI wavelet coefficients unit according to their importance using previously acquired ROI mask information.

In most applications, static ROI coding alone can be sufficient. During encoding processing, however, if any ROIs are not known, the ROI must be handled by decoding one. This is called dynamic ROI coding. The dynamic coding process extracts the basic coding information and the LL band's content and transfers them to the decoder from the not-coded and compressed bit stream. After the users view the decoded LL band's content, they specify their ROI, and the decoder transfers the information to the encoder. The encoder distinguishes either the ROI code block or ROI packet and constructs the last ROI mask. Subsequently, it assigns priorities according importance relevant to the ROI, handles ROI coding, and finally, transfers the reconstructed image to the decoder. To conduct an experiment, we compare it to Maxshift method, Modified Implicit method and Slope information based method [9].

The Maxshift method [7] is JPEG2000 part 1 standard. It divides the coefficients of ROI and background of quantized coefficients. It determines s, the highest value of the background coefficients, and then shifts the ROI coefficients on the bitplanes larger than s. Eq. 1 is an equation to obtain s, and Eq. 2 is an equation to obtain $a'(u,v)$, the coefficients after processing the ROI.

$$s \geq \max(M_b) \tag{1}$$

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

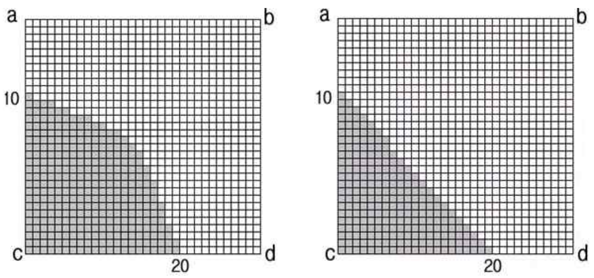
In Eq. 1, $max(M_b)$ means the biggest value of the quantized background coefficients in each subband. In Eq. 2, $M(u,v)$ is the ROI mask information. It is equal to 1 when the coordinate of the coefficient belongs to the ROI and 0 when one of the coefficients belongs to the background.

The Modified implicit method [10] is a method that complements the Implicit method [10] by reducing the priorities of the background coefficients included in the ROI code block without an algorithms complexity. Priority adjustments of the background coefficients in the ROI code block are conducted by truncating the background coefficients as k LSB. Eq. 3 transforms $a(u,v)$, which is the coefficient in the ROI code block before the ROI is processed to $\bar{a}(u,v)$, which are the coefficients after the priority adjustment.

$$\bar{a}(u,v) = \begin{cases} sign[a(u,v)] \left\lfloor \dfrac{|a(u,v)|}{2^k} \right\rfloor 2^k, & background\,coef. \\ a(u,v)\ , & ROI\,coef. \end{cases} \tag{3}$$

Parameter k is designated in the encoder and can be adjusted. The higher k, the lower the background coefficients priorities in the ROI code block are. The disadvantage is incompatible with the lossless ROI coding method due to the truncating of k LSB included in the ROI code block.

The Slope information based method [9] scans 4 pixels of the corners in one code block. Then, based on the information, it scans the edges from the corners to obtain the boundaries between the ROI and the background. Thus, this can reduce the time taken to generate the mask. The gray part of (a) in (Fig. 1) is the ROI specified by the user, and the white part is the background. (b) depicts the results obtained from the Slope information based method that divides the area drawn by a line, based on the two edge points. This method is an extremely fast way to generate the mask, but has lower precision than the other methods.



(a) original boundary (b) boundary division

(Fig. 1) An example dividing ROI and background of Sloped based method

In this paper, we propose a method that constructs the mask table with the eight points on four edges of a code block. The mask table is determined with one of 48 predefined mask patterns obtained from the values (0 or 1) of the eight points. This method helps quickly to create a mask table.

3. Generating ROI Masks using 8 points of the Edges

In this chapter, we describe 48 ROI patterns and the algorithm to get them from 8 edge points.

3.1 ROI Patterns

At first, the image transferred from the encoder is a rough image, such that, LL band, and the user specifies the user's preferred region after viewing the rough image. The region is called ROI and the information about the region is ROI information that will be transferred to the server. Like this, ROI coded image is preferentially transferred and is displayed to the user. During this processing, it is more important to handle this quickly than to specify the ROI precisely, although the image quality decreases [9]. This study uses the code blocks with 32x32 sizes, and may use 64x64 sizes as well. However, if the code block size is larger, the precision of the ROI decreases. When attempting to make the ROI mask specify the ROI area, if the precise area is needed, Implicit method [7] needs to be used. Actually, it is important to roughly define the ROI area, and to generate the mask in a short time period. This study uses previously defined 32x32 ROI masks and generates the rough masks. The mask patterns for this study are 48 items and illustrated in <Table 1>.

〈Table 1〉 Searching order with ROI distribution

a b c d	Main Pattern	e f g h	Pattern details (00 01 10 11)
0 0 0 0		0 0 0 0	background
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			exception
0 1 1 1		x x 1 1	
1 0 0 0		x x 0 0	
1 0 0 1			exception
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	

In Table 1, *a*, *b*, *c* and *d* are the vertices of each code block and *e*, *f*, *g* and *h* are the middle points of 4 edge sides. The position of each point is that in (a) of (Fig. 2). 1 indicates a pixel included in the ROI, and 0 is included in the background. *x* refers to a value not determined by the values, *e*, *f*, *g* and *h*. Two previously determined values are expressed with either 1 or 0, and two x values can be 00, 01, 10 or 11 which are the pattern details. The patterns shown in <Table 1> include 48 items. The black part of a pattern is an area of the ROI, and the white one is the background. When a block is completely ROI, *abcd* is 1111. In the case of complete background, *abcd* is 0000. *efgh* points are not evaluated for the above two cases. The block expressed with exception means that the ROI is separated by two areas, and this case is regarded as non-existing.

〈Table 2〉 Searching order with ROI distribution

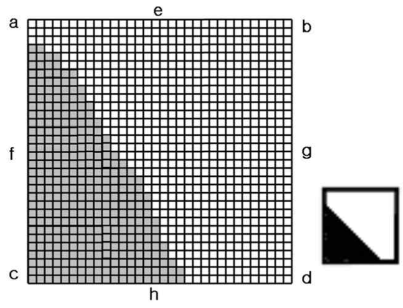
id (8 bits)		Pattern details(xx) (00 01 10 11)	conversion _tab()
a b c d	e f g h		
0 0 0 1	0 0 x x	16 17 18 19	1 2 3 4
0 0 1 0	0 x 0 x	32 33 36 37	5 6 7 8
0 0 1 1	0 x x 1	49 51 53 55	9 10 11 12
0 1 0 0	x 0 x 1	65 67 71 75	13 14 15 16
0 1 0 1	x 0 1 x	82 83 90 91	17 18 19 20
0 1 1 1	x x 1 1	115 119 123 127	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

3.2 Searching a Pattern

(a) of (Fig. 2) is a mixed code block of ROI and background in one block. The gray part indicates the ROI and the white part does the background. The pattern for (a) block is *abcd* = 0010 and *efgh* = 0101. Thus, (b) pattern is acquired. The black part of (b) is coded with ROI and the white part is coded with background. The following is an algorithm for searching a pattern.

```
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
```

As shown in (Fig. 2), the gray part of (a) is filled in with 1, and the white part is filled in with 0. All of the patterns consist of 48 items and the indexes from 1 to 48 are namely the pattern indexes. 0 means the pattern number which all pixels in a block are background, and 49 means one which they are all ROI. The patterns of 0 and 49 are determined without searching for *efgh*. The id consists of 8 bits, the 48 patterns get the values expressed in <Table 2> from <Table 1>. These values use a *conversion_tab()* function to coincide with the pattern numbers. For example, if a = 1, b = 0, c = 1 and d = 0, then f = 1 and g = 0. Here, if e = 0 and h = 1, then id = $10100101_{(2)}$ and id = $165_{(10)}$ as a details pattern $xx(eh)$ = 01. An id = 165 is chosen by *conversion_tab()* with a pattern number 30. The pattern number acquired by this process is sent to the encoder and used in ROI coding.

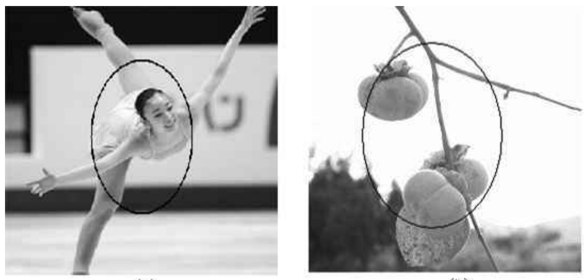


(a) (b)

(Fig. 2) (a) Sample code block and 8 points, (b) Its pattern

4. Experimental Results and Analysis

For experimentation we uses the Maxshift, Modified implicit, Slope based and the proposed method, and compare them using PSNR and the time to generate ROI masks. To ensure an unbiased experimental condition, we set k to 5. The images in (Fig. 3) for experimentation are Yuna and Persimmon. The shape of the ROI is an ellipse and the position is the center of the image. The sizes of the ROI are 25% and 30% of the full image respectively.


(a) (b)

(Fig. 3) Images used for experiments

4.1 Image Quality Evaluations

All of the coding methods offered in JPEG2000 standards conduct a sequential scan to generate the mask, and only the Slope based method scans a portion of the mask. However, the proposed method scans 6 points at most and significantly reduces the scan time because it uses the predefined mask patterns with the ROI mask. However, as shown in <Table 3> from <Table 4>, the proposed method has no choice but to be decreased in image quality compared to the other methods. The reason is why the method uses an approximate value for a quick ROI coding. The experiment evaluates the objective image quality using PSNR. <Table 3> and <Table 4> show the image qualities for Yuna and Persimmon in different methods, respectively.

〈Table 3〉 Comparisons of ROI PSNR in different methods using Yuna (bit rates=bpp)

Bit rates \ Coding methods	Maxshift	Modified implicit	Slope based	Proposed method
0.0625	28.031	24.845	24.616	24.199
0.125	30.495	28.567	28.340	27.930
0.25	34.067	32.699	32.557	32.341
0.5	39.453	38.480	38.343	37.914
1.0	47.301	46.900	46.349	45.955

〈Table 4〉 Comparisons of ROI PSNR in different methods using Persimmon (bit rates=bpp)

Bit rates \ Coding methods	Maxshift	Modified implicit	Slope based	Proposed method
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

4.2 Comparisons of time to generate ROI

The Slope based method and the proposed method only scan a portion of the mask information, as shown in <Table 5>. The Maxshift and Modified implicit methods scan all of the pixels, and the Implicit method scans all of the pixels only in the worst case. However, the Implicit method can complete scanning in one time regarded as the ROI block if a pixel is determined as the ROI at first scan. The Slope based method has the same scan time as the proposed method when the code block is either background or ROI. In the case of the mixed code block, the former three methods have a scan time that is proportion to n^2, while the Slope based method does to n. On the contrary, the proposed method has only 6 scan times, as it uses 4 points a, b, c, and d, and 2 points from e, f, g, and h.

In Maxshift and Modified methods, all the pixels in one block are scanned whether each pixel belongs to ROI or background. In Slope based and proposed methods, only 4 angular points are scanned while a block only includes all ROI pixels or all background pixels. But in case of mixed block, Slope method scans 4 angular points and n edge points, and the proposed method scans 4 angular points and each middle point on 2 edges.

〈Table 5〉 Mean Scan Times for different Methods

Scan times \ Coding methods	Maxshift	Modified method	Slope based	Proposed method
Background code block	n^2	n^2	4	4
ROI code block	n^2	n^2	4	4
Mixed code block	n^2	n^2	4 + n	6

〈Table 6〉 Mean Mask Generation Times for different Methods

Scan times \ Coding methods	Maxshift	Modified method	Slope based	Proposed method
Background code block	0	0	0	0
ROI code block	n^2	n^2	4	1
Mixed code block	n^2	n^2	$n^2/2$	1

<Table 6> depicts the time spent to generate an ROI mask. As the proposed method has predefined masks, the ROI mask is merely acquired to index the pattern number from the decoder. Alternatively, the other methods have to complete one additional process to generate the ROI mask using several expressions or bit masks. In the degree of correctness, the Maxshift and Modified implicit methods that scan all regions and represent the masks with each coefficient are superior to others. In Maxshift and Modified methods, all mask bits in one block are scanned like <Table 5>. In ROI code block, Slope based method only uses 4 angular points which means ROI block and in case of mixed block, this method touches half of one block on average. The proposed method just get the mask table from prepared masks at once. So, the proposed method reduced either the scan time of the mask bits or the masks generation time, but resulted in a decrease in image quality just a little compared to the other methods. In this paper, we evaluated subjective qualities of different methods but not objective quality. The reason is why it is hard for human eyes to discover a little difference of the image quality [8, 9]. As <Table 3> and <Table 4>, a little difference of PSNRs is meaningless to human eyes.

5. Conclusion

In the existing ROI methods, there must be an extensive scan process in order to generate the ROI mask. Because of this, it takes a considerable amount of time and the backgrounds around the ROI are regarded as ROI, and makes effective coding difficult. The Slope based method has reduced the time more than the existing methods, but takes time as before. However, the proposed method only scans the 8 edge points and then reduces the scan time. The generation time of the ROI mask using the pattern number acquired after scanning becomes reduced as well. As this method roughly generates the ROI mask, the image quality is poorer when compared with the existing methods, but this can be beneficial in real time processing. In the future research, we will apply this method to the dynamic ROI coding, utilizing an automatic ROI extraction.

References

[1] N. Kock, Media Richness or Media, IEEE Transactions on Profession Communication, Vol.48, No.3, 2005.

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

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

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

[5] 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, 2000.

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

[7] 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, 2000.

[8] K. Kang, A Fast and Dynamic Region-Of-Interest Coding Method Based on the Patterns in JPEG2000 Images, IJICIC, Vol.5, No.5, 2009.

[9] Y. Seo and K. Kang, A Slope Information based Fase Mask Generation Technique for ROI Coding in JPEG2000, IJICIC, Vol.6, No.6, pp.2817-2825, 2010.

[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, pp.590-596, 2005.

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

강 정 현

e-mail : ceo@ciclife.com
2002년 한국방송대학교 컴퓨터공학과(학사)
2008년 경상대학교 경영대학원 전자상거래학과(석사)
2011년 경상대학교 경영정보학과 박사수료
1995년～1997년 천령신문사 과장
1997년～2002년 하나로신문 부장
2003년～현 재 씨아이씨라이프(주) 상무이사
관심분야 : JPEG2000, 이러닝웹접근성, 자동안내방송

서 영 건

e-mail : young@gnu.ac.kr
1987년 경상대학교 전산과(학사)
1997년 숭실대학교 전산과(박사)
1989년～1992년 삼보컴퓨터
1997년～현 재 경상대학교 컴퓨터과학과 교수
2001년～현 재 경상대학교 공학연구원
2011년～현 재 University of North Carolina at Chapel Hill, School of Medicine, Visiting Scholar
관심분야 : Bioinformatics, Medical Imaging, Image segmentation, JPEG2000