A FAST IDENTIFICATION METHOD FOR JPEG 2000 IMAGES WITH DIFFERENT CODE-BLOCK SIZES AND RESOLUTION LEVELS

Hiroaki IWAI, Marielena PALACIOS PEREZ, and Hitoshi KIYA

Department of Information and Communication Systems Engineering
Tokyo Metropolitan University
6–6 Asahigaoka, Hino-shi, Tokyo 191–0065, Japan
E-mail: iwai-hiroaki@sd.tmu.ac.jp, palaciosperez-marielena@sd.tmu.ac.jp, kiya@sd.tmu.ac.jp

ABSTRACT

This paper proposes a method for identifying JPEG 2000 images with the different code-block sizes and the number of resolution levels. The proposed method is fast, because it utilizes the number of zero-bit-plane extracted without EBCOT decoding which requires lots of calculation time. Moreover, the proposed method does not produce false negative matches regardless of the compression rate, code-block size, and the resolution levels. The experimental results show the effectiveness of the proposed method.

Keywords: JPEG 2000, image identification, image search

1. INTRODUCTION

Recently, the use of digital images and video sequences is increased enormously because of the rapid growth of internet and multimedia systems. It is often necessary to identify an intended image in a database that has a large amount of digital images. The database images are generally stored in compressed forms to reduce the amount of data. JPEG 2000 [1], one of the modern compression standards, may become widely used in the near future, since it was officially selected as the standard compression technology for digital cinema [2]. Thus, it is crucial to develop methods for identifying JPEG 2000 images compressed from one original image.

The identification method for JPEG 2000 images using sign information of DWT (discrete wavelet transform) coefficients has been proposed [3]. This method needs lots of calculation time, since the entropy decoding called EBCOT (embedded block-based coding with optimized truncation) decoding is required to extract the sign information. On the other hand, the identification method using the number of zero-bit-planes has been proposed [4–6]. This method does not require EBCOT decoding, since the number of zero-bit-plane is extracted from the packet header. However, these methods cannot identify the images compressed with different code-block sizes and the number of resolution levels.

This paper proposes a method for identifying JPEG 2000 images with the different code-block sizes and the number of resolution levels. The proposed method runs fast, since the method utilizes the number of zero-bit-planes. In addition, the propose method does not produce false negative matches regardless of compression rate, the code-block size, and the number of resolution levels.

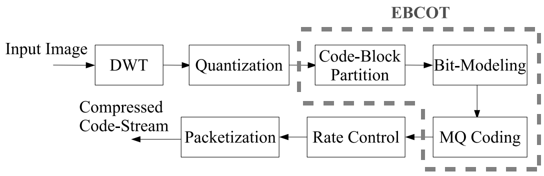

Fig. 1. Block diagram of JPEG 2000 encoder.

2. BACKGROUND

In this section, a brief overview of JPEG 2000 [1] is described, and the conventional identification methods using the feature of JPEG 2000 [3–6] are described.

2.1. Brief Overview of JPEG 2000

This section describes the outline of JPEG 2000 encoding, code-block and resolution level, bit-plane, and the structure of JPEG 2000 code-stream.

2.1.1. JPEG 2000 Encoding

Fig. 1 shows the block diagram of a JPEG 2000 encoder. DWT decomposites an input image into different resolution levels. These resolution levels contain a number of subbands, and the subbands are further divided into code-blocks as shown in Fig. 2. DWT coefficients in the code-blocks are quantized, and then are encoded by EBCOT. Rate control is performed to conform the amount of the code-stream to a target size. Finally, a JPEG 2000 compliant code-stream is generated by adding packet headers, main header, and some other control codes.

2.1.2. Code-Block and Resolution Level

Fig. 2 shows an example of image I analyzed by DWT. R^I denotes the number of resolution levels which is three in Fig. 2, and $r^I = 1, 2, \ldots, R^I$ is the index of the resolution level. b denotes the index of subband, where 1, 2, 3, and 4

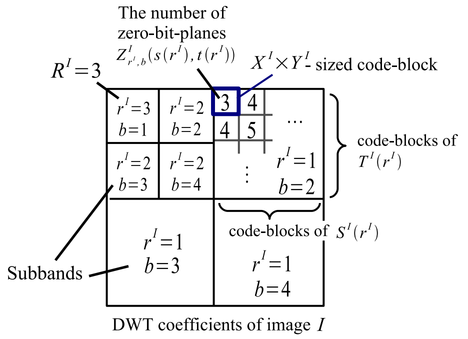

Fig. 2. Definition of subband, code-block, and resolution level.

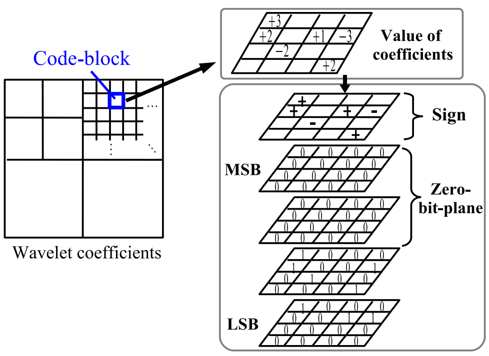

Fig. 3. Bit-plane decomposition.

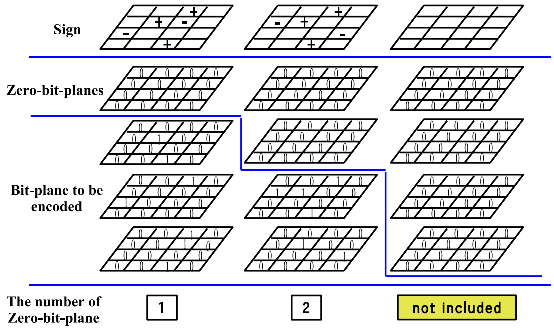

Fig. 4. Number of zero-bit-plane and "not included."

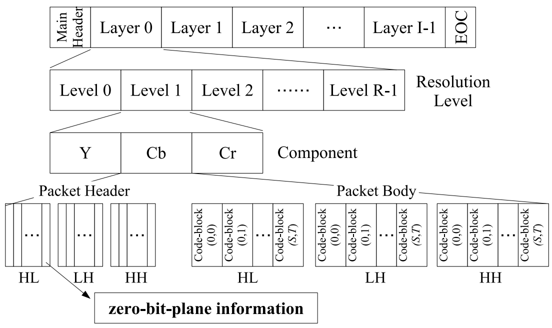

Fig. 5. Structure of JPEG 2000 code-stream.

are for LL, HL, LH, and HH, respectively. The subbands are divided into $X^I \times Y^I$-sized code-blocks, and $S^I\left(r^I\right) \times T^I\left(r^I\right)$ of code-blocks are in one subband of resolution level r^I.

2.1.3. *Bit-Plane*

Each quantized coefficient is separated into its sign and absolute magnitude as shown in Fig. 3, where the absolute magnitudes are factorized as bit-planes from the MSB (most significant bit) to the LSB (least significant bit). All of the samples in the bit-planes are either zero or one.

A special bit-plane, in which all samples are zero, frequently appears as shown in Fig. 3. This special bit-plane is referred to as "zero-bit-plane." When all bit-planes are zero-bit-plane as shown as the third code-block in Fig. 4, this code-block is defined as "not included," because the code-block does not include any data to be encoded. It is noted that the numbers of zero-bit-planes in images compressed with different quantization step sizes may be different, even if the compressed images are generated from one original image.

Hereafter, $Z^I_{r^I,b}(s,t)$ represents the number of zero-bit-planes in the code-block at (s,t), where $s = 1, 2, \ldots, S^I\left(r^I\right)$ and $t = 1, 2, \ldots, T^I\left(r^I\right)$.

2.1.4. *Structure of JPEG 2000 Code-Stream*

A JPEG 2000 code-stream consists of more than one packet. A packet has a packet header and a packet body, and the number of zero-bit-planes of each code-block which belongs to the packet is stored in the packet header as shown in Fig. 5. Therefore, the number of zero-bit-plane in a code-block can be obtained by only parsing the packet header, i.e., without heavy EBCOT decoding.

2.2. Conventional Methods and Those Problems

This section describes the conventional methods for identifying JPEG 2000 images [3–6]; one using sign information of DWT coefficients [3] and the other using the number of zero-bit-planes [4–6].

In the former method [3], the extracted sign information from query and database code-streams are compared. This method does not produce false negative matches regardless of the compression rate, code-block size, and the number of resolution level. However, this method needs lots of calculation time, because heavy EBCOT decoding is required to extract the sign information.

On the other hand, in the latter method [4–6], the number of zero-bit-plane extracted from the packet header are compared per code-block between the query and database images. The EBCOT decoding is not required in this method, so the required calculation time in these methods is smaller than the former method [3]. However, these methods cannot identify the images compressed by using either different code-block sizes or the number of resolution levels.

In the next section, a novel zero-bit-plane based identification method is proposed for JPEG 2000 images with different code-block sizes and resolution levels. The proposed method is fast and does not produce false negative matches

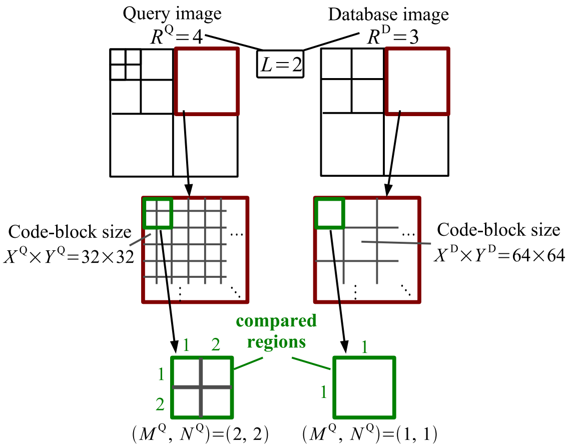

Fig. 6. Regions to be compared.

regardless of the compression rate, code-block size, and the resolution levels.

3. PROPOSED METHOD

This section proposes a method that identifies JPEG 2000 coded images with the different code-block sizes and the different number of resolution levels. The proposed method utilizes the number of zero-bit-planes that is obtained without heavy EBCOT decoding, so the method works fast.

3.1. Algorithms

The proposed method consists of three algorithms. The first algorithm defines regions to be compared. In the second algorithm, the numbers of zero-bit-planes in the compared regions are derived. The third algorithm compares the query and database images by using the numbers of zero-bit-planes in the regions.

It is assumed that query image Q has $X^{\mathrm{Q}} \times Y^{\mathrm{Q}}$-sized code-blocks and is decomposite to R^{Q} of resolution levels, whereas database image D has $X^{\mathrm{D}} \times Y^{\mathrm{D}}$-sized code-blocks and consists of R^{D} of resolution levels.

3.1.1. Definition of Regions to Be Compared

To normalize or unify the regions to be compared, this algorithm defines the size of compared regions and the compared depth of resolution levels as shown in Fig. 6. The algorithm is as follows:

1 By Eqs. (1) and (2), the numbers of code-blocks in a compared region are derived.

$$(M^{\mathrm{Q}}, M^{\mathrm{D}}) = \begin{cases} \left(\frac{X^{\mathrm{D}}}{X^{\mathrm{Q}}}, 1\right), & X^{\mathrm{Q}} \le X^{\mathrm{D}} \\ \left(1, \frac{X^{\mathrm{Q}}}{X^{\mathrm{D}}}\right), & \text{others} \end{cases}, \quad (1)$$

$$(N^{\mathrm{Q}}, N^{\mathrm{D}}) = \begin{cases} \left(\frac{Y^{\mathrm{D}}}{Y^{\mathrm{Q}}}, 1\right), & Y^{\mathrm{Q}} \le Y^{\mathrm{D}} \\ \left(1, \frac{Y^{\mathrm{Q}}}{Y^{\mathrm{D}}}\right), & \text{others} \end{cases}, \quad (2)$$

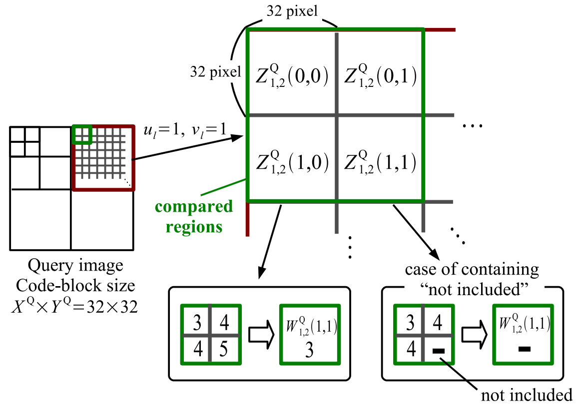

Fig. 7. The number of zero-bit-planes in compared regions.

where $M^{\mathrm{Q}} \times N^{\mathrm{Q}}$ of code-blocks form a compared region in query image Q and a compared region in database image consists of $M^{\mathrm{D}} \times N^{\mathrm{D}}$ of code-blocks.

2 The compared depth of resolution levels is given by Eq. (3).

$$L = \begin{cases} R^{\mathrm{Q}}, & R^{\mathrm{Q}} = R^{\mathrm{D}} \\ R^{\mathrm{Q}} - 1, & R^{\mathrm{Q}} < R^{\mathrm{D}} \\ R^{\mathrm{D}} - 1, & R^{\mathrm{Q}} > R^{\mathrm{D}} \end{cases}, \quad (3)$$

where L is the compared depth.

It is noted that the comparable depth decreases by one without decoding under the condition that $R^{\mathrm{Q}} \neq R^{\mathrm{D}}$. The number of the compared regions, c, is given as

$$c = \begin{cases} S^{\min}(L)T^{\min}(L) + \sum_{b=2}^{4}\sum_{l=1}^{L-1} S^{\min}(l)T^{\min}(l), & R^{\mathrm{Q}} = R^{\mathrm{D}} \\ \sum_{b=2}^{4}\sum_{l=1}^{L} S^{\min}(l)T^{\min}(l), & \text{others} \end{cases}, \quad (4)$$

where $S^{\min}(l) = \min_{I\in\{\mathrm{Q,D}\}} S^{I}(l)$ and $T^{\min}(l) = \min_{I\in\{\mathrm{Q,D}\}} T^{I}(l)$.

3.1.2. Derivation of the Numbers of Zero-Bit-Planes

This algorithm derives the numbers of zero-bit-planes in the all compared regions, i.e., in each compared regions in the subband b at l-th resolution level, where $b = 1, 2, 3$, and 4, and $l = 1, 2, \ldots, L$. The algorithm is as follows:

1 $l \leftarrow 1$.

2 $b \leftarrow 2$. If $R^{\mathrm{Q}} = R^{\mathrm{D}}$ and $l = L$, then $b \leftarrow 1$.

3 $u_l \leftarrow 1$.

4 $v_l \leftarrow 1$.

5 The number of zero-bit-planes at the (u_l, v_l) compared region in the b-th subband of the l-th resolution levels of image I, $W^{I}_{l,b}(u_l, v_l)$, is given as

$$W^{I}_{l,b}(u_l, v_l) = \min_{m^I, n^I} Z^{I}_{l,b}\left((u_l - 1)M^{I} + m^{I}, (v_l - 1)N^{I} + n^{I}\right), \quad (5)$$

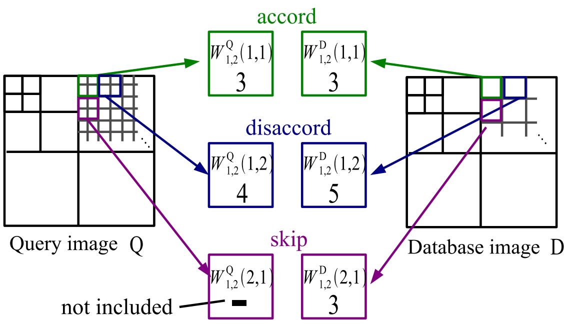

Fig. 8. The number of zero-bit-planes comparison for images having the same quantization step size.

where

$$m^I = 1, 2, \ldots, M^I, \qquad n^I = 1, 2, \ldots, N^I. \quad (6)$$

It is noted that Eq. (5) should satisfy

$$1 \leq (u_l - 1)M^I + m^I \leq S^I(l), \quad (7)$$
$$1 \leq (v_l - 1)N^I + n^I \leq T^I(l). \quad (8)$$

Note that if at least one of $Z^I_{l,b}\left((u_l-1)M^I + m^I, (v_l-1)N^I + n^I\right)$ among m^I's and n^I's is "not included," then $W^I_{l,b}(u_l, v_l)$ is also defined as "not included" as shown in Fig. 7.

6 $v_l \leftarrow v_l + 1$. If $v_l \leq T^{\min}(l)$, then go to Step 5.

7 $u_l \leftarrow u_l + 1$. If $u_l \leq S^{\min}(l)$, then go to Step 4.

8 $b \leftarrow b + 1$. If $b = 3$ or $b = 4$, then go to Step 3. Otherwise, go to Step 9.

9 $l \leftarrow l + 1$. If $l \leq L$, then go to Step 2. Otherwise, go to Step 10.

10 The numbers of zero-bit-planes among all c compared regions are obtained.

3.1.3. Image Identification

According to the quantization step condition, the proposed method switches two sub algorithms. That is, for images having the same quantization step size based on [4, 5], and for those having different sizes based on [6]. They are consequently described.

3.1.3.1. For the Same Quantization Step Size

The sub algorithm (Fig. 8) is as follows.

1 $l \leftarrow 1$.

2 $b \leftarrow 2$. If $R^Q = R^D$ and $l = L$, then $b \leftarrow 1$.

3 $u_l \leftarrow 1$.

4 $v_l \leftarrow 1$.

5 If either $W^Q_{l,b}(u_l, v_l)$ or $W^D_{l,b}(u_l, v_l)$ is "not included," then proceed to Step 7. Otherwise, then go to Step 6.

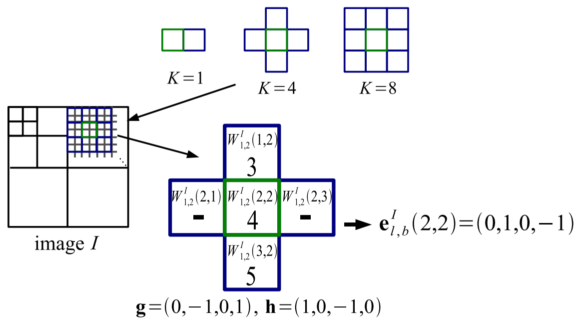

Fig. 9. Derivation of a relation for images having different quantization step sizes.

6 If $W^Q_{l,b}(u_l, v_l) = W^D_{l,b}(u_l, v_l)$, then proceed to Step 7. Otherwise, D is determinate that it is different from Q.

7 $v_l \leftarrow v_l + 1$. If $v_l \leq T^{\min}(l)$, then go to Step 5.

8 $u_l \leftarrow u_l + 1$. If $u_l \leq S^{\min}(l)$, then go to Step 4.

9 $b \leftarrow b + 1$. If $b = 3$ or $b = 4$, then go to Step 3. Otherwise, go to Step 10.

10 $l \leftarrow l + 1$. If $l \leq L$, then go to Step 2. Otherwise, D is identified as Q.

3.1.3.2. For Different Quantization Step Sizes

This sub algorithm utilizes the relation among the focused compared region and K of its neighboring regions to compare images as shown in Fig. 9. This sub algorithm consists of two parts: the relation derivation and the image identification.

The former part is as follows.

1 Define the neighboring regions by the coordinate difference. That is,

$$\mathbf{g} = (g_1, g_2, \ldots, g_K), \qquad \mathbf{h} = (h_1, h_2, \ldots, h_K), \quad (9)$$

where g_k and h_k indicate the horizontal and vertical distance between the focused region and the k-th neighboring region, respectively, c.f., Fig. 9.

2 $l \leftarrow 1$.

3 $b \leftarrow 2$. If $R^Q = R^D$ and $l = L$, then $b \leftarrow 1$.

4 $u_l \leftarrow 1$.

5 $v_l \leftarrow 1$.

6 $k \leftarrow 1$.

7 Relation between the focused and the k-th neighboring regions, $e^I_{l,b,k}$ is

$$e^I_{l,b,k}(u_l, v_l) = \begin{cases} 1, & W^I_{l,b}(u_l, v_l) - W^I_{l,b}(u_l + g_k, v_l + h_k) > 0 \\ 0, & W^I_{l,b}(u_l, v_l) - W^I_{l,b}(u_l + g_k, v_l + h_k) = 0 \\ -1, & W^I_{l,b}(u_l, v_l) - W^I_{l,b}(u_l + g_k, v_l + h_k) < 0 \end{cases} \quad (10)$$

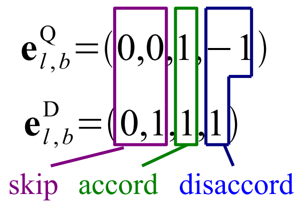

Fig. 10. Comparing relations between images having different quantization step sizes.

Note that if either $W^{I}_{l,b}(u_l, v_l)$ or $W^{I}_{l,b}(u_l + g_k, v_l + h_k)$ is "not included," and if either $u_l + g_k \leq 0$, $u_l + g_k > S^{\min}(l)$, $v_l + h_k \leq 0$, or $v_l + h_k > T^{\min}(l)$, then $e^{I}_{l,b,k} = 0$.

8 $\mathbf{e}^{I}_{l,b}(u_l, v_l) = \left(e^{I}_{l,b,1}(u_l, v_l), e^{I}_{l,b,2}(u_l, v_l), \ldots, e^{I}_{l,b,k}(u_l, v_l)\right)$ is derived. $k \leftarrow k + 1$. If $k \leq K$, then go to Step 7.

9 $v_l \leftarrow v_l + 1$. If $v_l \leq T^{\min}(l)$, then go to Step 6.

10 $u_l \leftarrow u_l + 1$. If $u_l \leq S^{\min}(l)$, then go to Step 5.

11 $b \leftarrow b + 1$. If $b = 3$ or $b = 4$, then go to Step 4. Otherwise, go to Step 12.

12 $l \leftarrow l + 1$. If $l \leq L$, then go to Step 3. Otherwise, go to Step 13.

13 The relations of c of compared regions are obtained.

The latter part (Fig. 10) is as follows:

1 $l \leftarrow 1$.

2 $b \leftarrow 2$. If $R^{Q} = R^{D}$ and $l = L$, then $b \leftarrow 1$.

3 $u_l \leftarrow 1$.

4 $v_l \leftarrow 1$.

5 $k \leftarrow 1$.

6 If $e^{Q}_{l,b,k}(u_l, v_l) \neq 0$ and $e^{D}_{l,b,k}(u_l, v_l) \neq 0$, then proceed to Step 7. Otherwise, go to Step 8.

7 If $e^{Q}_{l,b,k}(u_l, v_l) = e^{D}_{l,b,k}(u_l, v_l)$, then proceed Step 8. Otherwise, D is determined that it is different from Q.

8 $k \leftarrow k + 1$. If $k \leq K$, then go to Step 6.

9 $v_l \leftarrow v_l + 1$. If $v_l \leq T^{\min}(l)$, then go to Step 5.

10 $u_l \leftarrow u_l + 1$. If $u_l \leq S^{\min}(l)$, then go to Step 4.

11 $b \leftarrow b + 1$. If $b = 3$ or $b = 4$, then go to Step 3. Otherwise, go to Step 12.

12 $l \leftarrow l + 1$. If $l \leq L$, then go to Step 2. Otherwise, D is identified as Q.

3.2. Features

The proposed method is fast, because the number of zero-bit-planes can be extracted without EBCOT decoding. Moreover, the method does not produce false negative matches regardless of the differences in quantization step size, the code-block size, and the number of resolution levels between images. This section describes three points that let the proposed method have the latter feature.

3.2.1. The Numbers of Zero-Bit-Plane in Compared Regions

The sizes of code-blocks are defined by a power of two size in the standards [1, 2]. Thus, aggregating some small code-blocks forms a compared region that is comparable with a large code-block as shown in Fig. 6. Moreover, the number of zero-bit-planes in a compared regions consisting of small code-blocks is naturally given as the smallest number among the code-blocks as described in Sect. 2.1.3. These two consideration of JPEG 2000 coding characteristics let the proposed method avoid the false negative matches.

3.2.2. Compared Resolution Levels

In JPEG 2000 encoding, HL, LH, and HH subbands in the same resolution level from the same image correspond each other, since DWT is repeatedly performed for LL subband in each level. Thus, each subband in the resolution level from 1 to L, where L is derived from Step 2 in Sect. 3.1.1, are the same among images compressed from one original image. This consideration of JPEG 2000 characteristics let the proposed method avoid the false negative matches.

3.2.3. Two Identification Sub Algorithms

When quantization step sizes are the same between images, the numbers of zero-bit-planes in compared regions are directly compared in the proposed method as described in Sect. 3.1.3.1. On the other hand, when quantization step sizes are different between images, the algorithm described in Sect. 3.1.3.2 is used in the method, since the numbers of zero-bit-planes may change between the images because of quantization. The algorithm in Sect. 3.1.3.2 utilizes the feature that the distribution of the number of zero-bit-plane in the focused code-block and its neighboring code-blocks is retained regardless of quantization step size. Thus, the images with different quantization step sizes can be identified without the false negative matches in the proposed method.

4. EXPERIMENTAL RESULTS

In this section, the performance of the identification is demonstrated to verify the effectiveness of the proposed method.

4.1. Experimental Condition

In this experiment, frames from "Elephant Dream" are used. The number of frames are 1000 (from 3700th to 4699th

Table 1. Test images and JPEG 2000 encoding condition in database.

Test images	Elephant Dream 1000 images (3700–4699 frames)
Resolution	1,920(H) × 1,080(V)
Format	RGB, 8-bit precision/component
Resolution levels	5
code-block size	64 × 64
Base step size	0.005
Rate control	1.0bpp

Table 2. JPEG 2000 encoding condition of query image.

	Query 1	Query 2
Frame number	4250th frame	
Resolution levels	4	
code-block size	32 × 32	
Base step size	0.005 (same as database)	0.003
Rate control	1.0bpp	

frame), where each frame has $1,920 \times 1,080$ pixels. Tables 1 and 2 show JPEG 2000 encoding condition for constructing the database and compression parameters for the query images, respectively. The base step size in these tables is the parameter to derive quantization step sizes [1]. The query image is 4250th frame in Elephant Dream as shown in Fig. 11. The number of compared neighboring regions described in the algorithm of Sect. 3.1.3.2, K, is 8.

The proposed method is only compared with the DWT sign-based conventional method [3], because the other conventional method using the number of zero-bit-planes [4–6] is essentially impossible to identify the images compressed with different code-block sizes and the number of resolution levels.

4.2. Experimental Results

The result of the conventional method using the DWT signs [3] and the proposed method are shown in Table 3. There were no false negative matches in both methods. The conventional method did not produce false positive matches, whereas the proposed method produced false positives of 4 images. This is attributed to the fact that the numbers of zero-bit-planes in the false positive images are almost "not included,"

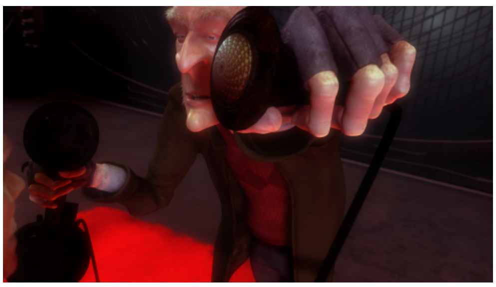

Fig. 11. Query image (4250th frame).

Table 3. Result of the conventional [3] and proposed method.

	Query 1		Query 2	
	conventional	proposed	conventional	proposed
Total identified images	1	5	1	5
Same original images	1	1	1	1
False positives	0	4	0	4
False negatives	0	0	0	0
execution time per image (msec)	550	7.68	546	22.2

since these images are fade-in and fade-out frames.

The execution time per image of the proposed method was about 72 times faster than the conventional method for query 1, and about 25 times faster for query 2.

5. CONCLUSIONS

This paper has proposed a novel zero-bit-plane based identification method for JPEG 2000 images with different code-block sizes and resolution levels. The proposed method is fast, since the method utilizes the extracted the number of zero-bit-plane without EBCOT decoding. Moreover, the method does not produce false negative matches regardless of the compression rate, code-block size, and the resolution levels.

REFERENCES

[1] ISO/IEC 15444-1:2004, "Information technology–JPEG 2000 image coding system–Part 1: Core coding system," Sep. 2004.

[2] "Digital Cinema system specification version 1.2," Digital Cinema Initiatives, LLC Member Representatives Committee, Mar. 2008.

[3] O. Watanabe, A. Kawana, and H. Kiya, "An identification method for JPEG 2000 images using the signs of DWT coefficients," Proc. IWAIT, no.P3–43, pp.846–850, Jan. 2007.

[4] O. Watanabe, T. Iida, T. Fukuhara, and H. Kiya, "Identification of JPEG 2000 images in encrypted domain for Digital Cinema," Proc. IEEE International Conference on Image Processing, no.MA.PJ.8, pp.2065–2068, Nov. 2009.

[5] T. Fukuhara, K. Hosaka, and H. Kiya, "Accurate identifying method of JPEG 2000 images for Digital Cinema," Proc. ACM/IEEE International Multimedia Modeling Conference, LNCS 4903, pp.380–390, Jan. 2008.

[6] T. Iida, T. Fukuhara, and H. Kiya, "A codestream-level identification method for JPEG 2000 coded images with different quantization step sizes," Proc. IEICE Signal Processing Symposium, no.C2-1, pp.392–397, Nov. 2008.