Codestream-Based Identification of JPEG 2000 Images with Different Coding Parameters

Osamu WATANABE†††, Takahiro FUKUHARA††, Members, and Hitoshi KIYA†††, Fellow

SUMMARY A method of identifying JPEG 2000 images with different coding parameters, such as code-block sizes, quantization-step sizes, and resolution levels, is presented. It does not produce false-negative matches regardless of different coding parameters (compression rate, code-block size, and discrete wavelet transform (DWT) resolutions levels) or quantization step sizes. This feature is not provided by conventional methods. Moreover, the proposed approach is fast because it uses the number of zero-bit-planes that can be extracted from the JPEG 2000 codestream by only parsing the header information without embedded block coding with optimized truncation (EBCOT) decoding. The experimental results revealed the effectiveness of image identification based on the new method.

key words: JPEG 2000, image identification, image search, digital cinema

1. Introduction

The use of digital images and video sequences has greatly increased recently because of the rapid growth of the Internet and multimedia systems. It is often necessary to identify a certain image in a database that has a large number of digital images in various types of applications such as editing and re-encoding in digital cinema applications. The perfect identification in such operations is defined as to find only the identical image for a given query image, however, the actual identification system is designed to extract candidate images. Therefore, the purpose of “identification” in this work is defined as the operation of finding an image that is identical to a given original image from an image database.

JPEG 2000 [11] has been officially selected as the standard compression/decompression technology for digital cinema by the Digital Cinema Initiatives consortium [12]. There is need to identify a certain frame in some operations such as editing and re-encoding in digital cinema applications. The perfect identification in such operations is defined as to find only the identical image for a given query image, however, the actual identification system is designed to extract candidate images. Therefore, the purpose of “identification” in this work is set to reduce the number of such candidate images. Moreover, the identification system used for digital cinema systems must be able to handle a large number of frames encoded with JPEG 2000 in a sufficiently short processing time.

Several methods have been developed for identifying compressed images [1], [2], [5]. The method described in Ref. [1] is for JPEG images and uses the signs of the discrete cosine transform (DCT) coefficients of the images. One method for JPEG 2000 [2], uses the signs of the discrete wavelet transform (DWT) coefficients. An algorithm for both JPEG 2000 and JPEG was also proposed [5]. Although these methods are for compressed images, they use transformed coefficients. This means that embedded block coding with optimized truncation (EBCOT) decoding is necessary to obtain the transformed coefficients for the identification process. Because EBCOT decoding is known as the most time consuming process in a JPEG 2000 decoder, the processing speed of the methods is not quite fast enough. However, codestream-based identification methods for JPEG 2000 images [3], [4] have also been proposed. As codestream-based identification means that there is no need for EBCOT decoding, such methods offer fast identification. However, these codestream-based methods assume that the original (or non-compressed) image of a query is always available and compressed images have the same JPEG 2000 coding parameters. Therefore, they do not work with prospective use cases where the original image is not available, or with the encoded images using different JPEG 2000 coding parameters, such as code-block sizes, DWT resolution levels, and quantization step sizes. This disadvantage is not trivial because the JPEG 2000 coding profiles for digital cinema defined in Ref. [14] may differ in coding parameters. The present authors have investigated techniques to solve this problem [15].

A method of unifying the differences in coding parameters are presented to identify JPEG 2000 images with different coding parameters in this paper. Image identification based on the proposed approach does not produce false-negative matches, regardless of differences in the coding parameters of JPEG 2000. Moreover, identification that is scalable can be carried out by using the scalability of the JPEG 2000 codestream. Simple binary comparison of the compressed data does not have these features. In addition, the new method is fast because it uses the number of zero-bit-planes, which is obtained by simply parsing the header part of a JPEG 2000 codestream. This paper is organized as follows. Section 2 provides a brief summary of the JPEG 2000 coding system, followed by the number of zero-bit-planes on which the proposed method is based. The notations and terminology used in this paper are also provided in

Copyright © 2012 The Institute of Electronics, Information and Communication Engineers
this section. Section 3 summarizes the conventional method of identifying images with JPEG 2000 and discusses problems with the method. Several techniques of the identifying JPEG 2000 images with different coding parameters are proposed in Sect. 4. Section 5 presents the experimental results of image identification based on the proposed approach. Section 6 is the conclusion.

2. JPEG 2000 Coding Technology

2.1 Overview

Figure 1 has a block diagram of the JPEG 2000 encoder. The coding procedure can briefly be summarized as follows. A discrete wavelet transform (DWT) analyzes the input image and generates subband domains, which are groups of wavelet coefficients. These coefficients are quantized and grouped in code-blocks. The value of each coefficient is represented in the bit-plane by a positive or negative sign and an absolute value. The embedded block coding with optimized truncation (EBCOT) scheme [13] is the entropy coder for JPEG 2000. It generates a compressed codestream by using a combination of bit-modeling and MQ coding (arithmetic coding). Some coding passes are truncated to keep the length of the codestream within a given target length for purposes of controlling the coding-rate. An incremental packet header is added to the codestream to make it JPEG 2000 compliant in the last stage of the encoder.

2.2 Bit-Plane Coding and Number of Zero-Bit-Planes

JPEG 2000 uses a bit-plane architecture that is completely different from other compression technologies such as JPEG and MPEG. Quantized coefficients are represented in sign-magnitude form and they have three dimensions of: horizontal, vertical, and bit-depth. The number of bit-planes is the same as the bit-depth, from most significant bits (MSBs) to least significant bits (LSBs). As outlined in Fig. 2, the sign bit-plane is at the MSB level, and the magnitude bit-planes are beneath it. The number of samples in a bit-plane is equal to that in the code-block, and all the samples in a bit-plane are either 0 or 1. The quantized magnitudes have a $K_{\text{max}}$-bit representation. The value of $K_{\text{max}}$ depends on the values of the quantization parameters, which may be different for each subband. The block coder for JPEG 2000 first determines the number of bits, $K \leq K_{\text{max}}$, that are needed to represent the quantized magnitudes. The encoder ideally finds the smallest such $K$. The difference between $K_{\text{max}} - K$ is defined as follows.

$$K_{\text{msbs}} = K_{\text{max}} - K$$

Here, $K_{\text{msbs}}$ represents the number of most significant magnitude bits that is skipped to encode with the encoder. The decoder will take this to be zero for all samples. This is called “the number of zero-bit-planes”. The remaining $K$ magnitude bits must be explicitly coded. Note that the value of $K_{\text{msbs}}$ is explicitly included as part of the code-block’s tag information in the packet header.

An example of a code-block with 16 samples (4 rows x 4 lines) is outlined in Fig. 3. The number of zero-bit-planes in the first code-block (left-hand side) is one, because the only MSB bit-plane is a zero bit-plane. The number of zero-bit-planes in the second one is two and four in the fourth one (right-hand side) in the same way. The third code-block is

![Fig. 2 Bit-plane decomposition and sign-magnitude representation of DWT coefficients in a code-block: A zero-bit-plane is a special bit-plane in which samples are all zeros. Zero-bit-planes are arranged from the MSB to the LSB level.](image)

![Fig. 3 Number of zero-bit-planes and “not included.”](image)
different in that all the bit-planes are zero-bit-planes. Such a code-block in the JPEG 2000 standard is defined as “not included” because the code-block does not have any data to be encoded. Since the coding-rate in JPEG 2000 is normally controlled by discarding MQ-encoded codestreams from LSB to MSB, there is fundamentally no effect from the number of zero-bit-planes even if the coding rate changes. However, the numbers of zero-bit-planes in images compressed with different quantization step sizes may differ, even if the compressed images are generated from one original image and the other coding parameters are the same.

The number of zero-bit-planes is part of the header information of JPEG 2000 codestreams. This information is easily obtained by parsing the main header of JPEG 2000. In other words, extracting the number of zero-bit-planes does not require full JPEG 2000 decoding, i.e., it does not require heavy EBCOT decoding.

2.3 Notation and Terminology

Unless otherwise stated, the following notation and terminology will apply to the rest of this paper.

- \( Q \) denotes images of the query and \( D \) denotes those of the database. Both are used as superscripts for the following parameters.
- \( R \) denotes the number of DWT levels. \( r = 0, 1, 2, \ldots \). \( R - 1 \) is used as an index. \( r = 0 \) means the highest frequency band and \( r = R - 1 \) is the index for the lowest one.
- Indices of subbands are written as \( b \). The number of \( bs \) is defined as,
  \[
  b = \begin{cases} 
  0 & \text{LL} \\
  1 & \text{HL} \\
  2 & \text{LH} \\
  3 & \text{HH} 
  \end{cases}
  \]
- \( X \) denotes the horizontal and \( Y \) denotes the vertical size of the code-block.
- \( J_1(r) \) and \( J_2(r) \) denote the horizontal and the vertical numbers of code-blocks in a subband of DWT level \( r \), respectively.
- Let \( B_{r,b}[j] \equiv B_{r,b}[j_1, j_2] \) denote the sequence of code-blocks belonging to the subband \( b \) of DWT-level \( r \), where \( j_1 \) and \( j_2 \) are the horizontal and vertical coordinates of a code-block so that \( 0 \leq j_1 < J_1(r) \) and \( 0 \leq j_2 < J_2(r) \).
- \( Z_{r,b}[j_1, j_2] \) represents the number of zero-bit-planes of \( B_{r,b}[j_1, j_2] \). That is, \( Z_{r,b}[j_1, j_2] = K^{null} \) at \( B_{r,b}[j_1, j_2] \). Let \( Z_{r,b}[j] \equiv Z_{r,b}[j_1, j_2] \) denote the sequence of the number of zero-bit-planes, where \( j \in [0, J_1(r)] \times [0, J_2(r)] \), of code-blocks in a subband.

Figure 4 shows an example of code-blocks, subbands, and DWT levels for an image.

\[ B_{0,1}[0, 0], Z_{0,1}[0, 0] = 3 \]

\[ J_1(r) \text{ code-blocks} \]

\[ J_2(r) \text{ code-blocks} \]

\[ \text{Quantized DWT coefficients of image} \]

Fig. 4 Definition of subband, code-block, and DWT-level, where number of DWT decomposition is \( R = 3 \). Thin solid line means boundaries of subbands and bold solid line means boundaries of code-blocks. Dashed lines mean boundaries of DWT levels.

3. Conventional Methods

3.1 Summary

Previously reported methods [3], [4] have assumed that the original (non-compressed) version of a query image is always available and the coding parameters used to encode database images are common within a database.

The procedure for these methods can be summarized as follows. The header parser is used to extract the JPEG 2000 coding parameters of database images and the extracted parameters are applied to generate the number of zero-bit-planes of the query image. Note that MQ encoding and bit allocation are not performed because only the number of zero-bit-planes is required to identify images.

The methods identify the matches between query image \( Q \) and database image \( D \) using the four rules described below. Note that the notation of “not included” is treated as a negative value in the following part of this paper.

- **Rule 0:** Image \( D \) is identified as matching image \( Q \) if any of the following rules are satisfied for all \( B_{r,b}^Q[J] \) and \( B_{r,b}^D[J] \).
  - **Rule 1:** \( B_{r,b}^D[j_1, j_2] \) is identified as matching \( B_{r,b}^Q[j_1, j_2] \) if \( Z_{r,b}^D[j_1, j_2] \) equals to \( Z_{r,b}^Q[j_1, j_2] \).
  - **Rule 2:** If \( Z_{r,b}^D[j_1, j_2] \) is “not included”, \( B_{r,b}^D[j_1, j_2] \) is identified as matching \( B_{r,b}^Q[j_1, j_2] \) regardless of the value of \( Z_{r,b}^Q[j_1, j_2] \).
  - **Rule 3:** If \( Z_{r,b}^Q[j_1, j_2] \) is “not included”, \( Z_{r,b}^Q[j_1, j_2] \) is explicitly counted through the encoding process. The quantity that is counted is re-defined as \( Z_{r,b}^Q[j_1, j_2] + \).
The success rate for identification is increased by this re-defining of the “not included” code-blocks. For example, the third code-block in Fig. 3 is defined as 5+ instead of “not included” because there are actually 5 zero-bit-planes in the code-block. However, if there were non-zero bit-planes under the truncation point before bit allocation, the number of bit-planes would be 5 or more. If \( r_D[j_1, j_2] \) is equal to or greater than 5, they are identified as being the same. (i.e., “5+” for \( B^Q \) is identified as being the same as “5” and more for \( B^D \)).

The compared depth and regions are defined to unify the difference in DWT levels and code-block sizes. The details on this step are as follows,

(A) Unifying difference in DWT levels

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

\[
L = \begin{cases} 
R^Q & \text{if } R^Q = R^D \\
R^Q - 1 & \text{if } R^Q < R^D \\
R^D - 1 & \text{if } R^Q > R^P,
\end{cases}
\]  

where \( L \) is the compared depth. For example, the compared depth with \( R^Q = 4 \) and \( R^D = 3 \) is visualized in Fig. 6. Note that the value for \( L \) decreases by one under the condition where \( R^Q \neq R^D \). Note that only code-blocks that are included inside the subbands within the compared depth are used in the rest of this step. The valid range of \( r \), which is the index for DWT levels, is \( 0 \leq r < L \) in the compared depth. This means that the subbands in lower DWT level (larger value of \( r \)) are excluded from the following identification process. The reason is described as follows. When \( R^Q \neq R^D \), the DWT coefficients of a code-block in lower subbands are also different from each other. Therefore, the number of zero-bit-planes of such a code-block are also different. In Fig. 6, the area of \( r^Q = 2 \) and \( r^D = 3 \) is equal to that of \( r^D = 2 \). Because the size of code-block is identical for a image, the same number of code-blocks are defined in the area. For the database image, the area of \( r^D = 2 \) is LL subband, however, there are HL, LH and HH subbands in the area of \( r^Q = 2 \). It is well known that the number of zero-bit-planes in LL subband is quit different from that in HL, LH or HH subbands. Thus, in case of \( R^Q \neq R^D \), the lower subbands are excluded from the identification process.

The proposed method consists of three parts. The first part defines the compared regions to unify the difference in DWT levels and code-block sizes. The numbers of zero-bit-planes in the compared regions are derived in the second part. The last part compares the query and database images by using the relation vector that is obtained with the numbers of zero-bit-planes derived in the second part.

4. Proposed Method

The proposed method consists of three parts. The first part defines the compared regions to unify the difference in DWT levels and code-block sizes. The numbers of zero-bit-planes in the compared regions are derived in the second part. The last part compares the query and database images by using the relation vector that is obtained with the numbers of zero-bit-planes derived in the second part.
Fig. 7 Examples of compared regions: the horizontal and vertical size of the regions are set to the least common multiple of $X_Q$ and $X_D$, and $Y_Q$ and $Y_D$. The number of code-blocks included inside a compared region is expressed as $M_Q \times N_Q$ or $M_D \times N_D$.

(B) Unifying difference of code-block sizes

The compared regions inside the compared depth are defined to normalize the difference in code-block sizes between a query image and a database image. As seen in Fig. 7, a compared region is defined as a square region having the dimension of $\text{lcm}(X_Q, X_D) \times \text{lcm}(Y_Q, Y_D)$. We can also see a compared region defined for the query image includes four code-blocks inside itself and that defined for the database image includes only one code-block. The number of code-blocks included inside a compared region is expressed as $M \times N$, where $M$ and $N$ are obtained by

\[
(M_Q, M_D) = \begin{cases} \frac{X_Q}{M_Q}, 1 & X_Q \leq X_D \\ (1, \frac{X_D}{M_D}) & \text{others} \end{cases}
\]

(3)

\[
(N_Q, N_D) = \begin{cases} \frac{Y_Q}{N_Q}, 1 & Y_Q \leq Y_D \\ (1, \frac{Y_D}{N_D}) & \text{others} \end{cases}
\]

(4)

Thus, $M_Q \times N_Q$ and $M_D \times N_D$ code-blocks are included inside the compared region for query image $Q$ and database image $D$.

Finally, the value of $K_1(r)$ and $K_2(r)$ are defined as Eq. (5) and a compared region within the subband is expressed as $C_{r,b}[k_1, k_2]$, where $k_1 = 0, 1, \ldots, K_1(r) - 1$ and $k_2 = 0, 1, \ldots, K_2(r) - 1$.

\[
(K_1(r), K_2(r)) = \begin{cases} \left( \frac{J_1^Q(r) - 1}{M_Q}, \frac{J_2^Q(r) - 1}{N_Q} \right) & \text{for query image} \\ \left( \frac{J_1^D(r) - 1}{M_D}, \frac{J_2^D(r) - 1}{N_D} \right) & \text{for database image} \end{cases}
\]

(5)

This coordinate normalizes the difference in the code-block sizes between a query image and a database image.

4.2 Derivation of Numbers of Zero-Bit-Planes

This step derives the number of zero-bit-planes in all compared regions, i.e., in all compared regions in subband $b$ at the $r$-th DWT level.

Let coordinate $[m, n]$ mean the location of a code-block in compared region $C_{r,b}[k_1, k_2]$, where $m = 0, 1, \ldots, M - 1$, $n = 0, 1, \ldots, N - 1$. $W_{r,b}[k_1, k_2]$, which is the number of zero-bit-planes of the compared region, is set to the minimum value of $Z_{r,b}[m, n]$, c.f., Fig. 8 (a). Note that the value of $W_{r,b}$ is set to “not included,” if at least one code-block in compared regions has a “not included” value, as shown in Fig. 8 (b).

4.3 Formation of Relation Values/vectors for Identification

Let $W_{r,b}[k_1, k_2]$ mean the number of zero-bit-planes of current compared region $C_{r,b}[k_1, k_2]$.

(A) For uniform quantization step size

If the uniform quantization step size is used for a query image and a database image, the value of $W_{r,b}[k_1, k_2]$ are directly compared with that of $W_{r,b}[k_1, k_2]$.

(B) For non-uniform quantization step size

If the non-uniform quantization step size is used for a query image and a database image, the vector $v_{r,b}[k_1, k_2]$ is defined for a compared region to unify the difference in quantization step sizes. As shown in Fig. 9, the vector is formed from the number of zero-bit-planes of the region’s eight immediate neighbors. The element of the vector, $v_i(i = 0, 1, \ldots, 7)$, is defined by Eq. (6). Note that any subtraction with “not included” results in the value of element $v_i = 0, i = 0, 1, \ldots, 7$.

\[
\begin{align*}
v_0 &= \text{sign}(W_{r,b}[k_1, k_2] - W_{r,b}[k_1 - 1, k_2]) \\
v_1 &= \text{sign}(W_{r,b}[k_1, k_2] - W_{r,b}[k_1 + 1, k_2])
\end{align*}
\]
Now, the decision to identify the compared regions is ready to be made. For each $k \in [0, K_1(r)] \times [0, K_2(r)]$, the decision $D_{r,b}[k]$ is derived by Eq. (7) or (8). If $D_{r,b}[k]$ is "true" for all $r$, $b$, and all elements of $k$, the query and database images are considered to be identical.

**A) For uniform quantization step size**

For the case where a query image and a database image have the same quantization step size,

$$
D_{r,b}[k] = \begin{cases} 
\text{true} & \text{if } W^Q_{r,b}[k] = W^D_{r,b}[k] \\
\text{false} & \text{otherwise}
\end{cases}
$$

**B) For non-uniform quantization step size**

For the case where the quantization step sizes are different between a query and a database image,

$$
D_{r,b}[k] = \begin{cases} 
\text{true} & \text{if } v^Q_{r,b,i}[k] \cdot v^D_{r,b,i}[k] \geq 0 \\
\text{false} & \text{otherwise}
\end{cases}
$$

where $v^Q_{r,b,i}[k]$ denotes the $i$th element of the vector $v^Q_{r,b}[k]$.

The product of $v^Q_{r,b,i}[k] \cdot v^D_{r,b,i}[k] \geq 0$ means that the sign of the elements of $v^Q_{r,b}[k]$ and $v^D_{r,b}[k]$ are the same and it also means that zero elements of the vectors are treated as "don’t care".

Note that the number of elements of $D_{r,b}[k]$ depends on the size of the images and the JPEG 2000 coding parameters. For example, a popular specification of an image for digital cinema $(4,096 \times 1,714, \text{RGB})$ with five level DWT and $32 \times 32$ code-block sizes gives 20,976 code-blocks. If the size of code-block is set to $64 \times 64$, the number of code-blocks is 5,424. In this example, the number of the compared regions, $\sum_k \sum_l \sum_r C_{r,b}[k]$, is 5,424. Thus the number of elements of $D_{r,b}[k]$ for all $r$ and $b$ is also 5,424. However, for fast processing, the identification processing should be finished at the first 'false' decision is obtained. Therefore, the number of decision to be made is not always equal to the number of elements of $D_{r,b}[k]$.

### 4.5 Features

The proposed method has three main features.

**A) No need to obtain original version of query image**

The proposed method uses query images in JPEG 2000 compressed format, so that it is suitable for applications where non-compressed versions of query images are not available.

**B) Robustness against differences in coding parameters**

The compared depth and the compared region described in Sect.4.1 and the number of zero-bit-planes derived from the compared region in Sect.4.2 normalize the difference in DWT levels and code-block sizes. The relation vector formed in Sect.4.3 unifies the difference in the quantization step size. In JPEG 2000 coding, these four parameters have influence over the number of zero-bit-planes altogether. Therefore, the proposed approach is considered to be robust against differences in these coding parameters.
(C) Fast processing

The number of zero-bit-planes and the JPEG 2000 coding parameters used in the encoding process are recorded in the main header of the codestream. This means that all the quantities required for the new method can be extracted by parsing the header without time consuming EBCOT decoding. Therefore, the time complexity for the proposed method is considered to be small. In addition, the amount of data in extracted quantities is considered to be much smaller than that in the whole codestream. Consequently, the space complexity for the proposed approach is predicted to not be large.

5. Experimental Results

The performance of the identification methods in terms of their precision and processing speed of the image identification were evaluated to verify the effectiveness of the proposed method.

5.1 Conditions and Procedure

The Standard Evaluation Material (StEM) [16] was used as test sets. The original StEM includes 17,239 frames. There are 47 shots that include dynamic/static movements, different conditions of lighting and different type of camera movement. Moreover, the staff credit follows the shots. The StEM is known as one of the de facto standard test set for digital cinema applications. Figure 10 shows the examples of frames in the StEM. For the following experiments, visually full black frames were rejected from the test sequence used in the experiments. As a result, the test sequence contained 14,964 frames. The specifications for the test sequences are summarized in Table 1.

First, sequences of JPEG 2000 compressed images to be used as query images were built. The encoding parameters for the sequences are listed in Table 2. Kakadu [17] version 6.4.1 was used as a JPEG 2000 codec. A query image was selected from one of the encoded sequences. Table 3 lists the parameters for database images. Five different databases were also built with Kakadu. In other words, five frame sets encoded with five different sets of JPEG 2000 coding parameters were built.

The “CB,” “DWT-lev,” and “Qstep” in Table 3 mean sequences encoded using different combinations of code-block sizes, DWT levels, and quantization base step sizes. The details on the combinations are described in the following.

**Table 1** Specifications for standard evaluation material.

| Parameter                | Value          |
|--------------------------|----------------|
| Number of frames         | 14,964         |
| Frame rate               | 24fps          |
| Spatial resolution       | 4,096 (H) x 1,740 (V) |
| Color format             | RGB(4:4:4) 12 bits/component |

**Table 2** JPEG 2000 encoding parameters for query images.

| Codec              | Kakadu version 6.4.1 |
|--------------------|----------------------|
| Coding-rate (VBR)  | 56 Mbps, 100 Mbps, 174 Mbps |
| DWT Filter         | 9 x 7                |
| DWT Level          | 5                    |
| Base step size of quantization | 1/256 |
| Code-block size    | 32 x 32             |
| Tile decomposition | No                   |

**Table 3** JPEG 2000 encoding parameters for database images.

| Name | CB64 | CB128 | DWT-lev | Qstep |
|------|------|-------|---------|-------|
| DWT level | 5 | 5 | 4 | 5 |
| Base step size | 1/256 | 1/256 | 1/256 | 1/200 |
| Code-block size | 64 x 64 | 128 x 32 | 32 x 32 | 32 x 32 |

Note that the base step size is the Kakadu parameter used for scalar quantization. As the actual quantization step size is determined based on the base step size, the change in the base step size means a change in the quantization step size. A smaller base step size generally yields better quality in the compressed image.

Identification experiments were carried out with the one-out-of-all rule for all possible combinations of query and database images. The number of combinations of query and database images was 14,964 x 14,964 in total. Figure 11 outlines the procedure for these experiments.

5.2 Identification Results and Its Remarks

Table 4 summarizes the number of true-positive (TP), false-positive (FP), true-negative (TN) and false-negative (FN) results obtained from the identification experiments for all database sequences. From this table, it can be confirmed that identification based on the proposed method could correctly identify the query frames regardless of the difference in JPEG 2000 coding parameters without any false-negative matches. No false-negative matches means...
that the proposed method never missed the database image which had the same original image as a query image. The number of false-positive matches can be discussed in terms of the false-positive-rate (FPR), which is calculated with Eq. (9).

\[
FPR = \frac{FP}{(FP + TN)}
\]  

(9)

The value for FPR of the proposed method is under 1.0% for all the database sequences. However, as the conventional method could not treat the difference in the code-block size and DWT levels, identification results were not available for CB64, CB128, and DWT-lev sequences. For Qstep sequences, a zero value for \(TP\) was obtained. This means the conventional method failed with identification.

The details of the number of \(FP\) of the proposed

---

Table 4 Number of true-positive (\(TP\)), false-positive (\(FP\)), true-negative (\(TN\)), and false-negative (\(FN\)) results obtained from identification experiment. The symbol “N/A” in the table means there were no available results because the conventional method could not be used for images with different coding parameters.

| Coding-rate | Database       | Proposed | Conventional [4] |
|-------------|----------------|----------|------------------|
|             |                | TP       | FP   | TN   | FN | TP | FP   | TN | FN |
| 56 Mbps     | CB64           | 14964    | 73487 | 223171454 | 0  | N/A | N/A  | N/A | N/A |
|             | CB128          | 14964    | 796241 | 223110891 | 0  | N/A | N/A  | N/A | N/A |
|             | DWT-lev        | 14964    | 708656 | 223197676 | 0  | N/A | N/A  | N/A | N/A |
|             | Qstep (4 neighbors) | 14964 | 74867 | 223157645 | 0  | 48365 | 223857967 | 14964 |
|             | Qstep (8 neighbors) | 14964 | 711871 | 223194461 | 0  | 48365 | 223857967 | 14964 |
| 100 Mbps    | CB64           | 14964    | 731339 | 223174993 | 0  | N/A | N/A  | N/A | N/A |
|             | CB128          | 14964    | 794139 | 223112193 | 0  | N/A | N/A  | N/A | N/A |
|             | DWT-lev        | 14964    | 504446 | 223401886 | 0  | N/A | N/A  | N/A | N/A |
|             | Qstep (4 neighbors) | 14964 | 737536 | 223168796 | 0  | 48365 | 223857967 | 14964 |
|             | Qstep (8 neighbors) | 14964 | 704176 | 223202156 | 0  | 48365 | 223857967 | 14964 |
| 174 Mbps    | CB64           | 14964    | 730651 | 223175681 | 0  | N/A | N/A  | N/A | N/A |
|             | CB128          | 14964    | 792715 | 223113641 | 0  | N/A | N/A  | N/A | N/A |
|             | DWT-lev        | 14964    | 504445 | 223401887 | 0  | N/A | N/A  | N/A | N/A |
|             | Qstep (4 neighbors) | 14964 | 736387 | 223169945 | 0  | 48365 | 223857967 | 14964 |
|             | Qstep (8 neighbors) | 14964 | 703580 | 223202752 | 0  | 48365 | 223857967 | 14964 |
method is described as follows. Actually, there are some query frames that returns quite large number of FP in the experiments. With careful investigation about the characteristic of such frames, it is found that the number of FP becomes large when the the most of code-blocks in a query frame have “not included” information because our method treats “not included” value as “Don’t care” to prevent omission of the TP frame. For frames with small code-blocks having “not included” information, average of the number of FP is small.

The $F_1$-score ($F_1$) \cite{18} is known to be one measure used in the field of information retrieval for measuring the performance of search, document classification, and query classification. A higher $F_1$-score means better performance. The value for the $F_1$-score is calculated as:

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$ (10)

$$\text{precision} = \frac{TP}{TP + FP}$$ (11)

$$\text{recall} = \frac{TP}{TP + FN}$$ (12)

Figure 12 plots the $F_1$-score calculated with the results, where the horizontal and vertical axes indicate the values for the bit-rates of query sequences and $F_1$-scores. From these results, it is confirmed that the proposed method does not produce any FN events. However, $F_1$-scores are not in enough high range because there are a lot of FP events. In this experiments, although the visually black frames were excluded, STEM has a lot of frames of the staff credits. Generally, frames of staff credits are very different from those of movie scene. The encoded frames of the staff credits have large number of “not included” code-blocks. For further discussion of the performance of the proposed method, the additional experiments were carried out. In the additional experiments, the frames of the staff credits were excluded and the number of frames for the experiments was 8,927. Without the number of frames, all conditions were the same as previous experiments. Table 5 shows the result of the additional experiments. From these results, it is confirmed that the proposed method scores higher range of $F_1$ values than those of the previous experiments.

The experiments described in the previous subsection were performed on a workstation with a Xeon 2.50 GHz processor and 4 GB of memory. The average processing time for the combination of a query and a database image was about 0.6 [ms/frame] excluding disk access time.

### Table 5

| Database   | TP    | TN    | FP    | FN    | $F_1$ score |
|------------|-------|-------|-------|-------|-------------|
| CB64       | 8927  | 79681723 | 679   | 0     | 0.963       |
| CB128      | 8927  | 79682002 | 400   | 0     | 0.978       |
| DWT-lev    | 8927  | 79644137 | 38255 | 0     | 0.318       |
| Qstep (4 neighbors) | 8927 | 79652598 | 29804 | 0     | 0.375       |
| Qstep (8 neighbors) | 8927 | 79678014 | 4388  | 0     | 0.803       |

### 6. Conclusion

A zero-bit-plane-based identification method for JPEG 2000 images with different JPEG 2000 coding parameters has been presented in this paper. New techniques such as compared depths, compared regions and relation vectors have been introduced to normalize the difference in DWT levels, code-block sizes, and the quantization step sizes between a query image and a database image. All the quantities required for the proposed method can be extracted by only parsing the header without time consuming EBCOT decoding and the amount of data used is small. Therefore, the time and space complexity with the proposed method is considered to be small. The experimental results confirmed that the proposed method never produced FN matches or decreased the accuracy of image identification.

### References

[1] F. Arnia, I. Iizuka, M. Fujiyoshi, and H. Kiya, “Fast and robust identification methods for JPEG images with various compression ratios,” Proc. ICASSP 2006, pp.397–400, May 2006.
[2] O. Watanabe, A. Kawana, and H. Kiya, “An identification method for JPEG2000 images using the signs of DWT coefficients,” IEICE Technical Report, IE2006-217, Jan. 2007.
[3] T. Fukuhara, K. Hosaka, and H. Kiya, “Accurate identifying method of JPEG2000 images for digital cinema,” Proc. 14th International Multimedia Modeling Conference (MMM’08), pp.380–390, Lect. Notes Comput. Sci., Jan. 2008.
[4] T. Fukuhara, K. Hosaka, and H. Kiya, “Identifying method of JPEG2000 images in the codestream level for digital cinema,” IEICE Trans. Inf. & Syst. (Japanese Edition), vol.91-D, no.9, pp.2205–2213, Sept. 2008.
[5] K.O. Cheng, N.F. Law, and W.C. Siu, “A fast approach for identifying similar features in retrieval of JPEG and JPEG 2000 images,” Proc. APSIPA ASC 2009, Oct. 2009.
[6] Y. Uchida and S. Sakazawa, “D-12-93 Near-Duplicate Video Detection Considering Temporal Burstiness of Local Features,” Proc. IEICE Gen. Conf. 2011, D-12-93, March 2011 (in Japanese).
[7] “Information technology – JPESearch – Part 1: System framework and components.” International Standard ISO/IEC TR 24800-1, Dec. 2007.
[8] “Compact descriptors for visual search: Applications and use scenarios.” ISO/IEC JTC1/SC29/WG11/N11529, July 2010.
[9] “Compact descriptors for visual search: Context and objectives.” ISO/IEC JTC1/SC29/WG11/N11530, July 2010.
[10] “Compact descriptors for visual search: Requirements.” ISO/IEC JTC1/SC29/WG11/N11531, July 2010.
[11] “Information technology — JPEG 2000 image coding system — Part 1: Core coding system.” International Standard ISO/IEC IS-15444-1, Dec. 2000.
[12] “Digital cinema system specification V1.2.” Digital Cinema Initiatives, LLC Technology Committee, March 2008.
[13] D. Taubman, “High performance scalable image compression with EBCOT,” IEEE Trans. Image Process., vol.9, no.7, pp.1158–1170, July 2000.
[14] “Information technology — JPEG 2000 image coding: Core coding system AMENDMENT 2: Extended profiled for cinema and video production and archival applications.” ISO/IEC 15444-1:2004/Am2, 2009.
[15] O. Watanabe, T. Fukuhara, and H. Kiya, “Fast Identification of JPEG 2000 images for digital cinema profiles,” Proc. ICASSP 2011,
Osamu Watanabe received his B.S., M.S., and Ph.D. degrees from Tokyo Metropolitan University in 1999, 2001, and 2004, respectively. In 2004, He joined the faculty of Takushoku University as a research associate in electronics and computer systems. Since 2008, He has been an Assistant Professor in the Department of Electronics and Computer Systems. He is a member of IEEE, ITE, and ISO/IEC/JTC1/SG29/WG1 committee “Joint Photographic Experts Group” (JPEG). His areas of research interest are image processing, image coding, and signal processing.

Takahiro Fukuhara received his B.S. and M.S. degrees from Waseda University in 1986 and 1988, respectively. He received the Ph.D. degree from Tokyo Metropolitan University in 2009. From 1988–1997 he was with Mitsubishi Electric Corp., Communication Systems Labs. In 2007, he joined Sony Corporation. He is a member of ISO/IEC/JTC1/SG29/WG1 committee, chair and editor of JPEG2000 Part1 AMD-1, AMD-2, co-editor of JPEG2000 Part3 (Motion-JPEG2000), and DCI (Digital Cinema Initiative) Standardization. He is a recipient of IEICE young engineer award (1992) and IPSJ contribution award to international standardization (2007). His areas of research interest are still/video coding and transmission, image identification and analysis, digital cinema.

Hitoshi Kiya received his B.E and M.E. degrees from Nagaoka University of Technology, Japan, in 1980 and 1982 respectively, and a Dr. Eng. degree from Tokyo Metropolitan University in 1987. In 1982, he joined Tokyo Metropolitan University, where he is currently a Professor of Information and Communication Systems, Faculty of System Design. He was a Visiting Fellow at the University of Sydney, Australia from Oct. 1995 to Mar. 1996. His research interests are in the areas of multirate systems and image processing, including filter bank design and theory, image and video coding, compressed-domain video manipulation and security for multimedia. He received the Telecommunications Advancement Foundation Award in 2011, the IEICE Engineering Science Society Contribution Award in 2010 and the IEICE Best Paper Award in 2008. He serves as an associate editor for IEICE Trans. Fundamentals, IEEE Trans. Signal Processing and IEEE TransImage Processing. He currently serves as the President of the IEICE ESS and a Vice President of the APSIPA. He is a Fellow of the ITE, a senior member of the IEEE.