FAST METHOD FOR JOINT RETRIEVAL AND IDENTIFICATION OF JPEG CODED IMAGES BASED ON DCT SIGN

Fitri ARNIA, Ikue IIZUKA, Masaaki FUJIYOSHI, and Hitoshi KIYA

Department of Electrical Engineering, Tokyo Metropolitan University
6-6, Asahigaoka, Hino City, Tokyo, 191–0065, Japan.

ABSTRACT

We propose a fast method to retrieve images from a JPEG image database. The proposed method is intended to retrieve similar images, including their compressed versions, and to identify exact match and all compressed versions of a query image simultaneously. Similarity level is measured based on the non-zero DCT coefficients signs, which serve as features. The method is simple and fast because the DCT coefficients signs can be obtained by only entropy-decoding the bitstream. There is no need to calculate features explicitly. Furthermore, the method is robust to JPEG compression, particularly for image identification purpose.

***Index Terms*—** DCT coefficient sign, image similarity, image identification, image retrieval, copy detection

1. INTRODUCTION

A creation of a large amount of digital images in the last decades has led to a more challenging problem of image retrieval. For fast retrieving of the images that are mostly saved in JPEG (Joint Photographic Image Group) compressed format, image retrieval should be accomplished in the compressed domain, in which a complete decoding is not required. The retrieval method should also be robust to JPEG compression. In this context, the method should fulfill the following conditions: (1) it can retrieve similar images, including all of their copies with different JPEG compression ratios and (2) it can identify the exact match and all compressed versions of a query image. The second condition addresses two of the CBIR (Content Based Image Retrieval) challenges stated by The Benchathlon Network for the search of a specific image [1].

Several image retrieval methods in the compressed domain have been previously proposed in [2–9]. The methods described in [2–4] use the DCT (Discrete Cosine Transform) coefficients signs as image features, which can be obtained directly by entropy decoding the bitstream. Thus their computational complexity is low. They [2–4] aim at identifying the exact match and all compressed versions of a query image (the second condition). However, method in [2, 3] is not able to retrieve similar images. While the method in [4] cannot perfectly identify all compressed version of a query image, resulting in false positive (FP) and false negative (FN).

Other methods are mainly designed to acquire similar JPEG [5–8] and JPEG 2000 images [9]. Most of the methods in the JPEG compressed domain calculate features explicitly [5–8]. Therefore, their computational complexity is higher than that of the method in [2–4]. While it was reported that the schemes are able to retrieve similar images, there was no extensive report on robustness of the schemes to JPEG compression.

In this paper, we proposed a technique that is able to retrieving similar images, including their compressed versions, and simultaneously identifying the exact match, including all compressed versions of a query image. That is, the method fulfills the robustness requirements described in the first paragraph. The robustness can be achieved due to the method utilizes features that is invariant to JPEG compression, namely the non-zero DCT coefficients signs. In addition, the method requires much lower computational complexity compare to that of, for example the method in [5] , and depending on the number of DCT coefficient taken into consideration, can be lower than that of the method in [4]. This can be achieved because the non-zero DCT coefficients signs can be obtained by only entropy decoding the bitstream. There is no requirement to do explicit calculations to obtain the features. Because the method is fast, it is suitable for real time image retrieval from a large database. Due to its robustness, the method can be useful for content-based copy detection.

(a) Original Image (b) DSOI

Fig. 1. Image of DCT Coefficients Signs.

2. DCT COEFFICIENTS SIGNS

The DCT coefficients signs of an image contain significant amount of information [4]. DCT coefficients signs only image (DSOI) preserves information such as outline and edges of original image. Figure 1 illustrates this condition. Figure 1 (a) is the original image and (b) is the DSOI of image (a). We can recognize the original image by observing its DSOI. Therefore, similar images may be retrieved by considering only the DCT coefficients signs.

For JPEG images, the DCT coefficients signs can be obtained by entropy decoding the JPEG bitstream. Figure 2 shows JPEG bitstream structure. Start of Image (SOI) marker is the first marker in the JPEG stream. It is followed by header, tables, Huffman code, and eventually ended by End of Image (EOI) marker. The Huffman code contains the compressed image data. The compressed data can be regarded as a collection of DCT coefficients, including signs and magnitudes. Each DCT coefficient sign can be obtained by following procedures. The decoder determines run/size of a given segment of a bitstream by searching the Huffman tables. Knowing the run/size, the next step is to obtain coefficient exact value (sign and amplitude), which is defined as additional bit. Positive and negative signs are represented as two's complement-related number [10]. For ex-

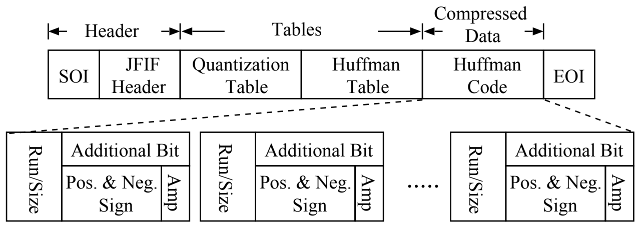


Fig. 2. JPEG Bitstream.

ample, if bit "0" precedes a segment of bits, it represents a negative sign, and vice versa. Therefore, positive and negative signs can be extracted by only Huffman (run/size) decoding, followed by signs evaluation.

3. PROPOSED METHOD

3.1. Model of Retrieval from JPEG Image Database

Model of image retrieval system from a JPEG database is shown in Fig. 3. Depending on the query images, we consider three cases, namely querying with:

1. Images with identical compression format (in this case JPEG) and identical size.
2. Images with different compression format and may be different size (we select JPEG 2000 image).
3. Others, which include raw image format.

For querying with JPEG images (case 1), it is only required to partially decode the JPEG bitstream, as explained in Sect. 2. For querying with different compression format (case 2), JPEG 2000 images is taken into consideration due to its spreading use and increasing popularity. In JPEG 2000 format, several thumbnail image can be obtained using its scalability feature. Querying with JPEG 2000 image is started by pre-processing the image. The processes are illustrated in dashed line in Fig. 3. For querying with the same image sizes, the query image is first decoded, then the 8×8 DCT is applied to the decoded image, and the DCT coefficients signs $\mathbf{S}_Q$ can be extracted. For querying with a JPEG 2000 thumbnail image, interpolation is conducted prior to applying the DCT. If the query is a raw image (case 3), the DCT coefficients signs $\mathbf{S}_Q$ can be extracted following calculation of the DCT coefficients. In all kind of querying, the DCT coefficients signs of JPEG image from database **D** are extracted in the same way as the JPEG query image. Similarity between $\mathbf{S}_Q$ and $\mathbf{S}_D$ is calculated finally.

3.2. Notations and Terminologies

Several notations and terminologies that are used in the following sections are listed here.

- S_Q and S_D represent the DCT coefficients signs obtained from query image Q and database image D respectively.
- M represents the number of constituent 8×8 block of an image. $0 \leq m \leq M-1$.
- N represents position of DCT coefficient according to the zigzag scanning in each 8×8 block. $0 \leq N \leq 63$.
- $\zeta_{Q,D}(m)$ is a counter of non-zero DCT coefficients signs of image Q and D in the mth block.
- $\chi_{Q,D}(m)$ is a counter of the same DCT coefficients signs of image Q and D in the mth block.

3.3. Retrieval Algorithm

The retrieval method is accomplished according to the following steps:

1. Set $m := 0$.

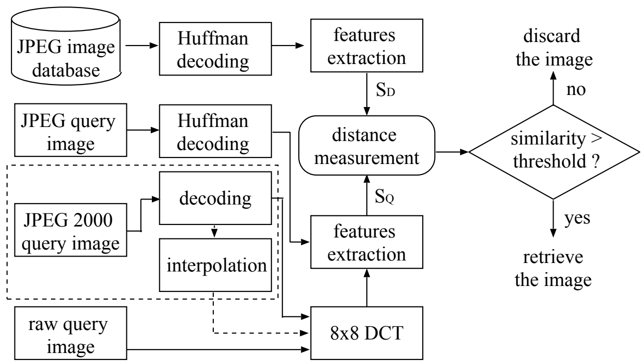


Fig. 3. Model of image retrieval from a JPEG image database.

2. Set $\zeta_{Q,D}(m) := 0$, $\chi_{Q,D}(m) := 0$ and $n := 0$.
3. If $S_Q(m,n) \neq 0$ and $S_D(m,n) \neq 0$, $\zeta_{Q,D}(m) := \zeta_{Q,D}(m) + 1$. If $S_Q(m,n) = 0$ or $S_D(m,n) = 0$, proceed to step 5
4. If $S_Q(m,n) = S_D(m,n)$, $\chi_{Q,D}(m) := \chi_{Q,D}(m) + 1$. If $S_Q(m,n) \neq S_D(m,n)$, $\chi_{Q,D}(m) := \chi_{Q,D}(m) - 1$.
5. Set $n := n+1$. If $n \leq 63$, proceed to step 3
6. Normalize $\chi_{Q,D}(m)$ with respect to $\zeta_{Q,D}(m)$.

$$r_{Q,D}(m) = \frac{\chi_{Q,D}(m)}{\zeta_{Q,D}(m)} \tag{1}$$

here, if $\zeta_{Q,D}(m) := 0$, then set $r_{Q,D}(m) := 1$, and $m := m+1$. If $m \leq M-1$, proceed to step 2.

7. Take average $\sigma_{Q,D}$ of $r_{Q,D}(m)$ with respect to M.

$$\sigma_{Q,D} = \frac{1}{M} \sum_{m=0}^{M-1} r_{Q,D}(m) \tag{2}$$

$\sigma_{Q,D}$ is defined as a similarity level between image Q and D. That is $\sigma_{Q,D}$ in Eq. 2 is the proposed similarity measure.

All the above-mentioned processes are applied to all images in the database **D** to obtain their similarity level. Actually, $\chi_{Q,D}(m)$ which is determined in step 4 can be calculated by following equation:

$$\chi_{Q,D} = \sum_{n=0}^{N-1} S_Q(m,n) \cdot S_D(m,n) \tag{3}$$

That is the total number of DCT coefficients signs product of query image Q and database image D. There are only two possible values of sign product, namely "1" and "-1". If both $S_Q(m,n)$ and $S_D(m,n)$ are the same, their product will be "1", and if $S_Q(m,n)$ is different from $S_D(m,n)$, their product will be "-1". This is in accordance with step 4.

3.4. Similarity based on DCT Coefficients Signs

The similarity of two images is measured based on the count of equivalent DCT coefficients signs of the images, which are tiled into 8×8 blocks. Equation 3 (or step 4 in Sect. 3.3) calculates similarity level of two 8×8-DCT blocks. Each pair of equivalent DCT coefficients signs contributes to similarity increment and vice versa. Similarity of the images is obtained by averaging the similarity of all 8×8 blocks. Note that zero valued coefficients are excluded prior to similarity calculation (implemented in step 3 in Sect. 3.3). For two similar images, several of their DCT coefficients signs are not equivalent and the number of non-equivalent signs increase as the images become less similar. For images that are compressed from the same original image with different compression ratios, all signs in the images are equivalent ($\chi_{Q,D} = \zeta_{Q,D}$). Condition in which $\chi_{Q,D} = \zeta_{Q,D}$ is fulfilled is referred to as identification and considered as a special case in the proposed method.

Table 1. Computational Complexity. M is total number of 8×8 block in an image.

a. Feature Extraction (FE)

Operation	Proposed	Method in [5]	Method in [4]	Pixel Corr.
$\pm$	-	$9\times M$	-	-
No.of Trans.	-	M	-	-

b. Similarity Measurement (SM)

Operation	Proposed	Method in [5]	Method in [4]	Pixel Corr.
$\pm$	M	$M+256$	$64\times M$	$448\times M$
$\oplus$	-	$M+256$	-	-
$\div/\times$	$M+1$	-	$64\times M$	$192\times M$

3.5. Computational Complexity

Calculations required to retrieve an image, which is commonly referred to as computational complexity includes feature extraction (FE) calculation and similarity measurement (SM). We compared computational complexity of the proposed method with the ones in [4, 5] and pixel correlation respectively. Method in [5] was selected to represent a method in the JPEG compressed domain that does not use directly the DCT coefficients signs as features. We consider pixel correlation for comparison purpose because it is a simple and widely-acceptable similarity measure in pixel domain. We use a general correlation formulation in [2]. Table 1 (a) and (b) show computational complexity for FE and SM respectively. For pixel correlation, calculations required to do a complete decoding is not shown in the table. For SM of the method in [4], calculations shown in Table 1 is the calculation required if all the DCT coefficients in an 8×8 blocks (64 coefficients) is considered. The uses of less than 64 DCT coefficients are also considered by the method in [4].

3.6. Robustness to JPEG Compression

Robustness to JPEG compression can be achieved due to the proposed method uses invariant features, that is the non zero DCT coefficients signs. The DCT coefficients signs are not changeable due to JPEG compression. Scalar quantization that is applied in the JPEG compression changes the number of zero coefficients, but does not change the DCT coefficients signs.

4. SIMULATION

Four methods were evaluated. They are the proposed method, method in [4], method in [5], and pixel correlation respectively. The method in [5] and the pixel correlation were selected based on the same reason as previously stated in sect. 3.5.

Three kinds of simulation were conducted for the purposes of (1) evaluating performance of the proposed method with respect to others in retrieving similar images, (2) testing robustness of the proposed method to JPEG compression, and (3) testing similarity level when JPEG image database is queried with other compression format, which in this case is JPEG 2000 image.

The metrics used to measure similarity performance are *recall* and *precision* (RP). Let n_c and n_m be the number of correct, and missed candidates respectively among the first M retrievals. The *recall* for query image q is defined as:

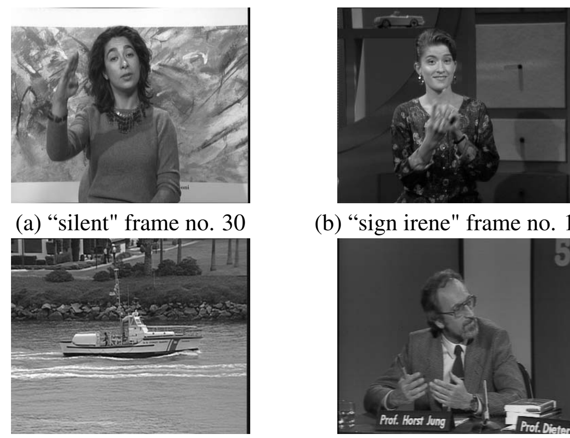
$$Recall_q = \frac{n_c}{n_c + n_m} \quad (4)$$

Let n_c and n_f be the number of correctly retrieved images and wrongly retrieve ones respectively among the first M retrievals. The *precision* for query image q is defined as:

$$Precision_q = \frac{n_c}{n_c + n_f} \quad (5)$$

Table 2. Image database. Each QF is composed of 1000 images.

Database	gray scale, 352 × 288, 8 bpp
class 1	silent (frame no. 1-100)
class 2	sign irene (frame no. 101-200)
class 3	hall monitor (frame no. 201-300)
class 4	coastguard(frame no. 301-400)
class 5	container(frame no. 401-500)
class 6	diskus(frame no. 501-600)
class 7	foreman(frame no. 601-700)
class 8	bowing(frame no. 701-800)
class 9	paris(frame no. 801-900)
class 10	deadline(frame no. 901-1000)
Q-factor	50, 200, 350



(a) “silent" frame no. 30 (b) “sign irene" frame no. 130

(c). "coastguard", frame no. 330 (d). "diskus", frame no. 470

Fig. 4. Example of frames used as queries.

A perfect identification performance is considered when the methods rank the exact match and all compressed versions of a query image in the highest rank, with the similarity value equal to "1" or distance value equal to "0", without false negative (FN) and false positive (FP).

4.1. Simulation Conditions

Image database used in the simulations was a collection of ten individual video sequences, whose details are summarized in Table 2. The sequences comprise of different kinds of scenery, which include camera operations such as zooming ("diskus") and camera panning ("coastguard" and "diskus") . Each sequence is referred to as a class, in which the similar images are grouped together. Each class is represented by 2 queries. Totally there are 20 queries, each with compression ratios (Q-factors (QFs)) 50. All frames were then JPEG compressed with three different QFs , 50, 200 and 350 respectively. Figure 4 shows four examples of query images.

The methods under evaluation use similarity measure to decide similar images. However, method in [5] uses distance measure. In similarity methods, the most similar image (exact match) will have similarity level equal to “1”. Conversely, in distance methods, the most similar image (exact match) will have distance equal to “0.”

4.2. Results of Similar Image Retrieval

In similarity evaluation, only images with QF 50 (1000 images), were considered. The recalls/precisions (RPs) values averaged over 20 queries with Q-factor 50, and $M = 100$ are shown in Table 3 (a). As we assumed that the frames in one class are very similar, we set $M = 100$ (RPs value will be equal to "1" if all frames from the corresponding class are retrieved). The performances of the proposed method are very closed to other methods. It is worth noting that

Table 3. Average of recall and precision.

(a). Query : 20 images, QF 50. Total image in database 1000

	Proposed	Method in [4]	Method in [5]	Pixel Corr.
Recall	0.95	0.99	0.96	0.95
Precision	0.95	0.99	0.96	0.95

(b). Query : 20 images, QFs 50, 200, 350. Total image in database 3000

	Proposed	Method in [4]	Method in [5]	Pixel Corr.
Recall	0.92	0.98	0.94	0.95
Precision	0.92	0.98	0.94	0.95

methods in [5] and pixel correlation need a much higher computational complexity (longer processing time). As an illustration, querying time (1 query) of the proposed method and the method in [5], were approximately 2.18 seconds and 642,13 seconds, respectively, when using a machine with CPU of 3.6 GHz and RAM of 2 GB.

4.3. Results of Robustness to JPEG Compression

For evaluation of robustness to JPEG compression, all 3000 images in the database (QFs 50, 200 and 350) were taken into account. Since there are 100 similar images from each class, which means 300 similar images with three different QFs, M was set to 300. Table 3 (b) shows the RPs values averaged over 20 query images. In general, average RPs are decreased because of the characteristics of camera-zoomed sequence ("diskus").

Table 4 (a) shows identification results of querying "sign irene", frame no. 130 with QF 50 (abbreviated as "sign irene 130-50"). It can be seen that all versions of the query image (sign irene, frame no. 130, QFs 50, 200 and 350 respectively) were perfectly retrieved with similarity value of "1", and were ranked as the first three highest ranks. Note that simulations with worse QFs, namely 400, 450 and 500 also produced similarity value of "1". The same results were achieved for all other query images.

On the other hand, methods in [4, 5] were not able to identify all compressed versions of the query image. For example, method in [4], put "sign irene 130-200" at rank 40 instead of rank 2 (see Table 4 b). In this case, false positive (FP) and false negative (FN) occurred. The pixel correlation could identify all compressed versions of the query image. However, even with the highest computational complexity, the similarity values of the other compressed versions are not equal to "1".

Table 5 summarizes the performances of all methods in terms of computational complexity, similarity and identification robustness to JPEG compression.

4.4. Results of Querying with Different Compression Format

For querying with different image format, we used the JPEG 2000 images. Due to the limited space, the results of querying can not be presented. However, the results are in line with those of querying with JPEG images.

5. CONCLUSION

A fast image retrieval method for a JPEG database is proposed. The method is able to retrieve similar images, including their compressed versions, and to identify exact matches and compressed versions of the query images simultaneously. Similarity of the images is measured based on the non-zero DCT coefficients signs, which serve as features. The proposed method is simple and fast because the features can be obtained by only entropy decoding the bitstream, without requirement of explicit calculations. The method is robust to JPEG compression, particularly for image identification purpose.

Table 4. Results of image identification with different QFs.

(a). Proposed

Rank	Retrieved Image	Similarity Value	Note.
1	sign irene130-50	1	identified
2	sign irene 130-200	1	identified
3	sign irene 130-350	1	identified

(b). Method in [4]

Rank	Retrieved Image	Similarity Value	Note.
1	sign irene 130-50	1	identified
2	*sign irene 131-50*	*0.7652*	*FP*
3	*sign irene 129-50*	*0.7404*	*FP*
40	*sign irene 130-200*	*0.4418*	*FN*
118	*sign irene 130-350*	*0.3071*	*FN*

(c). Method in [5]

Rank	Retrieved Image	Distance Value	Note.
1	sign irene 130-50	0	identified
2	*sign irene 131-50*	*98*	*FP*
3	*sign irene 129-50*	*102*	*FP*
20	*sign irene 130-200*	*220*	*FN*
53	*sign irene 130-350*	*278*	*FN*

(d). Pixel Correlation

Rank	Retrieved Image	Similarity Value	Note.
1	sign irene 130-50	1	identified
2	sign irene130-200	0.9779	identified
3	sign irene 130-350	0.9638	identified

Table 5. Performance summary. PC = pixel correlation, CC = computational complexity, Sim = similarity, Ident = identification.

	Proposed	Method in [4]	Method in [5]	PC
CC	lowest	low	higher	highest
Sim.	acceptable	good	good	good
Ident.	perfect	not perfect	not perfect	not perfect

6. REFERENCES

[1] The Benchathlon Network: CBIR Challenges - http://www.benchathlon.net/resources/challenges.html

[2] F. Arnia, I. Iizuka, M. Fujiyoshi and H. Kiya, "Fast image identification methods for JPEG images with different compression ratios," IEICE Trans. Fund. of Elect., Comm. and Comp. Sci., vol.E89-A, no.6,pp.1585-1593, June 2006.

[3] Fitri Arnia, I. Iizuka, M. Fujiyoshi and H. Kiya, "Fast and robust identification methods for JPEG images with various compression ratios," Proc. 2006 ICASSP, vol.2 , pp.397–400, May 2006.

[4] J. Bracamonte, M. Ansorge, F. Pellandini and P.-A. Farine, "Efficient compressed domain target image search and retrieval," Proc. CIVR 2005, LNCS3568, pp.154-163, 2005.

[5] C.-C. Chang, J.-C. Chuang and Y.-S. Hu, "Retrieving digital images from JPEG compressed image database," J. Image and Vision Computing, vol.22, pp.471–484, 2004.

[6] M. Shneier and M. Abdel-Mottaleb,"Exploiting the JPEG compression scheme for image retrieval," IEEE Trans. Pattern Anal. Mach. Intell., vol.18, no.8, pp.849–853, Aug. 1996.

[7] J. Jiang, A. Armstrong, and G.C. Feng "Web-based image indexing and retrieval in JPEG compressed domain," Multimedia System, vol.9, pp.424–432, 2004.

[8] A.R. McIntyre and M.I. Heywood,"Exploring content-based image indexing techniques in compressed domain," Proc. IEEE Canadian Conf. Elect. & Comp. Eng., vol.2, pp.957–962, May 2002.

[9] M.K. Mandal and C. Liu, "Efficient Image Indexing Techniques in the JPEG2000 Domain," J. Elect. Imaging, vol.13, no.1, pp.182–187, Jan. 2004.

[10] G.K. Wallace, "The JPEG still picture compression standard," IEEE Trans. Cons. Elect., vol.38, no.1, pp.xviii–xxxiv, Feb. 1992.