

RESEARCH

Open Access



Improved reversible data hiding in JPEG images based on new coefficient selection strategy

Fisseha Teju Wedaj¹, Suah Kim², Hyoung Joong Kim^{2*} and Fangjun Huang^{3,4}

Abstract

Recently, reversible data hiding (RDH) techniques for JPEG images have become more extensively used to combine image and authentication information conveniently into one file. Although embedding data in JPEG image degrades visual quality and increases file size, it is proven to be useful for data communication and image authentication. In this paper, a data hiding method in JPEG image using a new coefficient selection technique is proposed. The proposed scheme embeds data using the histogram shifting (HS) method. According to the number of zero AC coefficients, block ordering is used to embed data first in the blocks causing less distortion. In order to further reduce the distortion, positions of AC coefficients are selected carefully. Finally, AC coefficients valued +1 and -1 are used for embedding, and the remaining non-zero AC coefficients are shifted to the left or right directions according to their sign. Compared to the current state-of-the-art method, experimental results show that the proposed method has higher peak signal to noise ratio (PSNR) and smaller file size.

Keywords: Reversible data hiding, JPEG, AC coefficient selection

1 Introduction

These days, the growth of multimedia technologies and the attractiveness of the Internet are dramatically increasing. Multimedia technologies are key players in this digital information age. Data communication and information exchange between people are done in digital form and mostly over the open Internet. This means information exchange allows a third party to access all types of multimedia information. Easy accessibility of multimedia information threatens privacy, and there is no guarantee for multimedia ownership and integrity. In general, the reliability, security, integrity, and confidentiality of multimedia information are under risk while in digital form, especially on the Internet. Other than these risks, there are cases where important metadata such as encrypted patient information or digital signature for a file is accidentally deleted or the image itself is maliciously doctored. Reversible data hiding (RDH) provides a mitigation methodology for those types of scenarios by

allowing users to hide a payload into their cover media. It does it in a way that the original cover image can be recovered without any distortion. For some applications such as medical and military imaging, where even the slightest distortion is not desired, RDH can be useful.

JPEG standard is one of the oldest and most commonly used digital image formats in daily life. Most current media broadcasting corporations and digital devices use JPEG image compression to store information in graphic form. Data hiding in a JPEG compression domain in a reversible manner is also a useful and reasonable research area for image archive management, image authentication, and image privacy.

The first data hiding method was proposed by Barton's patent [1] in 1997. Following that, numerous schemes of data hiding and lossless data hiding have been proposed. Tian [2] proposed a difference expansion technique to embed data in the spatial domain; the host image is divided into pixel pairs, and the difference value of the two pixels in a pair is expanded to carry one message bit. Subsequently, Tian's work was improved upon in many aspects [3, 4]. In 2006, Ni et al. [5] proposed a histogram shifting technique to embed data more

* Correspondence: khj-@korea.ac.kr

²Graduate School of Information Security, Korea University, Seoul, South Korea

Full list of author information is available at the end of the article

efficiently, preferring to embed a message into coefficients belonging to some selected frequencies. The minimum points of the histogram are used for data embedding. Qu et al. [6] proposed a novel embedding strategy for reversible watermarking based on compensation. Some of the modified pixel values return to their original values after data hiding, compensating image distortion. These days, many algorithms exploit prediction errors (PE) and pixel value ordering (PVO) [4, 7–9]. Some authors [8, 9] use block-based pixel ordering, whereas one author [7] uses pixel-based ordering (neighboring pixel ordering). Sachnev et al. [4] proposed a popular efficient data hiding using a sorting and prediction technique. Among these research ideas, most use a histogram shifting (HS) strategy to embed data. In the HS method, while the embeddable bins have the payload embedded, the rest of the bins must be shifted either to the right or to the left depending on their sign. Embeddable and expandable bins are specified by the encoder threshold values. Compared to other methods, PE and PVO schemes have better embedding performance. However, the PVO method works well for low-capacity embedding. On the other hand, reversible data hiding based on least squared prediction [10–12] gained popularity. These methods work quite well in high-capacity embedding. Most of the existing reversible data hiding techniques focus on the pixel domain.

However, these days, researchers have been giving attention to data hiding in JPEG-compressed image. JPEG image compression is based on discrete cosine transform (DCT), which is one of the basic building blocks for JPEG compression [13, 14]. The most important aspect of DCT for JPEG compression is the ability to quantize the DCT coefficients using visually weighted quantization table values. In [15, 16], Huffman code mapping is used for embedding the payload into a JPEG bit-stream. They used the unused variable-length code (VLC) for AC coefficients by applying a map from the unused codes to used codes. Hu et al. [17] improved the VLC-based lossless data hiding scheme for JPEG images. In their work, a lossless data hiding scheme that directly embeds data into the bit-stream of JPEG images based on unused variable-length codes in the Huffman table is presented. Their method [17] is an improvement on the first method [15] of this kind. Chang et al. [18] presented a block-based lossless and reversible data hiding scheme for hiding payload in DCT-based compressed images. From each block of the medium-frequency elements, two successive zero coefficients are used for embedding. Mobasseri et al. [16] embedded payload in the JPEG bit-stream by code mapping.

The rest of this paper is organized as follows. In Section 2, an overview of the JPEG image standard is provided and related works of reversible data hiding in JPEG images are discussed. In Section 3, the proposed RDH scheme for

JPEG images is introduced. Experimental results and analysis are given in Section 4. Finally, Section 5 concludes the work.

2 Related works

2.1 Overview of legacy JPEG image compression standard

Joint Photographic Experts Group (JPEG) ([13, 14], <http://www.iijg.org/>) has worked on the very first international digital image compression standard. It is one of the most common image formats used by current digital cameras and other image-capturing devices. The compression ratio of the legacy JPEG compression standard is high with minimal loss in visual quality. The key steps of the JPEG compression process ([13, 14], <http://www.iijg.org/>) are shown in Fig. 1. The JPEG legacy encoder consists of three parts: discrete cosine transform (DCT), quantization step, and entropy encoding. The original image is divided into non-overlapping 8×8 pixel blocks, subtracted by 127 for normalization, transformed using the two-dimensional DCT function, and entropy coding is applied using the Huffman table. In the quantization step, DCT coefficients are divided according to the quantization table and rounded to the nearest integer. JPEG standard does not specify the DCT quantization table, but it provides a recommended table in their Annex. This means that anyone can use their own quantization table by scaling the table elements using different quality factors [13, 14]. When a different quantization table is used, different image qualities and compression ratios are achieved. The recommended quantization table is shown in Fig. 2. Quality factor (QF) is a value ranging from 1 to 100, and scaled quantization table Q_s is obtained using the following equation:

$$SF = \begin{cases} \frac{5000}{QF} & \text{if } QF < 50 \\ 200 - (2 \times QF) & \text{if } QF \geq 50 \end{cases} \quad (1)$$

$$Q_s(i, j) = \max \left\{ \left[Q(i, j) \times \frac{SF}{100} \right], 1 \right\} \quad (2)$$

where Q is the recommended quantization table value, and $[x]$ represents the round operator of x .

2.2 Reversible data hiding in JPEG image

Unlike the pixel-based reversible data hiding technique, JPEG reversible data hiding embeds in the quantized DCT coefficients. Reversible data hiding in JPEG DCT coefficients is based on four general data hiding approaches [19]. The first one is a lossless compression-based method proposed by Fridrich and Goljan [20], in which the embedding space is preserved by compressing the redundant component of the image. Since the message

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

a Recommended quantization table by JPEG

10	7	6	10	14	4	31	37
7	7	8	11	16	35	36	33
8	8	10	14	24	34	41	34
8	10	13	17	31	52	48	37
11	13	22	34	41	65	62	46
14	21	33	38	49	62	68	55
29	38	47	52	62	73	72	61
43	55	57	59	67	60	62	59

b Scaled $QF = 70$ quantization table

Fig. 1 Block diagram of the legacy JPEG compression

capacity is too small, this method has received less attention.

The second is a quantization table modification approach proposed by Fridrich et al. [21] and later improved by Wang et al. [22]. Their techniques work by preprocessing the quantized DCT coefficients and modifying the quantization table to create space for data hiding. Although the experimental results of [22] achieve high peak signal to noise ratio (PSNR), the file size increases greatly.

The third method [16] modifies the Huffman table. In this method, data embedding is performed by mapping a used variable length coding (VLC) to an unused VLC. Qian and Zhang [15] improved this method, but the payload size is very small for these methods.

The fourth method [18, 23] modifies the quantized DCT coefficients. Xuan et al. [23] shifted the quantized DCT coefficient histogram with an optimum searching strategy. This optimum strategy helped the technique to achieve good performance. In order to make data embedding unperceivable and the visual quality of marked image high, when a certain amount of data is embedded, only lower and middle frequency DCT coefficients are chosen to embed data in the embedding process. Sakai et al. [24] improved this scheme, producing better image quality with a new block selection strategy. Li et al. [25] proposed a reversible data hiding scheme on JPEG images based on the smaller DCT value selection method and three slight modifications of the quantization table.

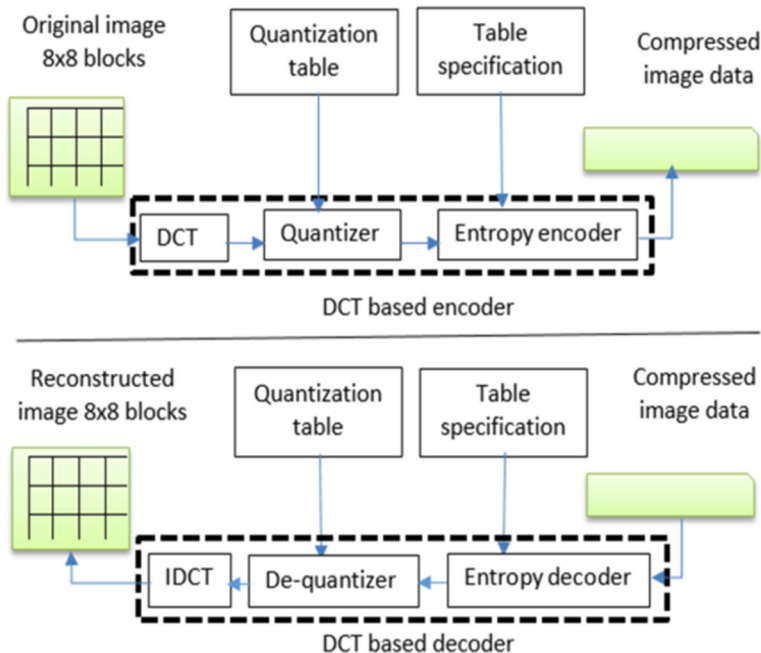


Fig. 2 Examples of JPEG quantization tables

Using the HS analogy, Lin et al. [26] discussed high-capacity reversible data hiding for JPEG images. They used different block sizes and got high embedding capacity. Celik et al. [27] also proposed lossless data hiding in the least significant bit (LSB) of the JPEG DCT coefficients. Those DCT values, which have high distortion, are padded as side information in the payload. Huang et al. [19] proposed a histogram shifting technique on AC coefficients valued +1 or -1 and employ a block ordering based on the statistical properties of the number of zero AC coefficients in the blocks. The method is quite fresh and achieves huge improvements compared to the past work.

3 Proposed scheme

This section describes the proposed JPEG reversible data hiding method. In subsection 3.1, the embedding of the AC coefficients will be explained. Subsection 3.2 explains the block ordering method proposed by Huang et al. [19] in details. Subsection 3.3 explains the main contribution of this paper, which is the process of selecting the positions of AC coefficients for minimizing the distortion. Subsection 3.4 details which side information is needed and how they are embedded. Subsection 3.5 briefly explains the recovery of the original DCT coefficients and the extraction of the payload. A short explanation of the complexity of the algorithm is also found in subsection 3.6. Finally, the section concludes with a precise description of the proposed algorithm pseudo code.

3.1 Embedding

In a quantized coefficient block, the very first coefficient is referred to as the DC coefficient, whereas 63 others are referred to as AC coefficients. The proposed method will only embed the payload in the AC coefficients. For each AC coefficient C , the following method is used to embed the payload bit $b \in 0, 1$ and creates the watermarked AC coefficient C' :

$$C' = \begin{cases} C + b, & \text{if } C = 1 \\ C - b, & \text{if } C = -1 \\ C + 1, & \text{if } C > 1 \\ C - 1, & \text{if } C < -1 \\ C, & \text{else} \end{cases} \quad (3)$$

A bit b is embedded if and only if C is either +1 or -1. If C is greater than +1 or less than -1, it is shifted by +1 and -1, respectively. Notice that the zero coefficients are not modified as explained in the beginning of this section. Before we continue with the rest of the explanation, the following histogram shifting terms are summarized for clarity:

- Embeddable coefficients: AC coefficients valued either +1 or -1.
- Unchangeable coefficients: AC coefficients valued 0.
- Shiftable coefficients: AC coefficients which are greater than +1 or less than -1.

3.2 Block ordering

Huang et al. [19] first proposed a block ordering based on the number of zero AC coefficients. The experimental results show that blocks with many zero AC coefficients will likely contain many -1 or +1 valued AC coefficients. Using this statistical feature, [19] proposed only embedding in +1 and -1 and set embedding order such that the blocks with more zero AC coefficients are embedded first. This strategy effectively reduced distortion. Additionally, the file size increase is also lessened. The modification of zero AC coefficient increases the file size. This is because whenever a zero AC coefficient is modified to non-zero, an extra symbol is needed to be coded. Therefore, their method leads to smaller distortion and smaller file size than the existing schemes, which embed in zero AC coefficients.

Before the AC coefficients selection step, in the proposed scheme, a similar block ordering scenario is used. A block with a higher number of zero coefficients will be at the top with the highest embedding priority and a block with less number of zero coefficients will be at the bottom with lowest embedding priority.

3.3 AC coefficient position selection

(1) Motivation: Block ordering method proposed by Huang et al. [19] has been proven to be effective for decreasing the distortion. However, they did not consider two additional points. First, each position of the AC coefficients across all blocks has different distribution of embeddable, unchangeable and shiftable coefficients. There are certain AC positions, which have more embeddable coefficients than shiftable coefficients. Figure 3 shows the total number of embeddable and shiftable coefficients in each position for all the blocks. Figure 3a is the statistics recovered from image Lena with $QF = 70$ and Fig. 3b is from image Baboon with $QF = 70$. For example, in Fig. 3a, it shows that positions such as 24 has a larger number of embeddable AC coefficients than shiftable AC coefficients. On the contrary, position 5 has smaller number of embeddable AC coefficients while having a larger number of shiftable AC coefficients. Naturally, position 24 is more desirable for embedding. Second, the modifying cost for each position is not uniform; i.e., due to the quantization table's non-uniform structure, modifying the AC coefficient which is near the DC coefficient may lead to less distortion than modifying the last one. For these

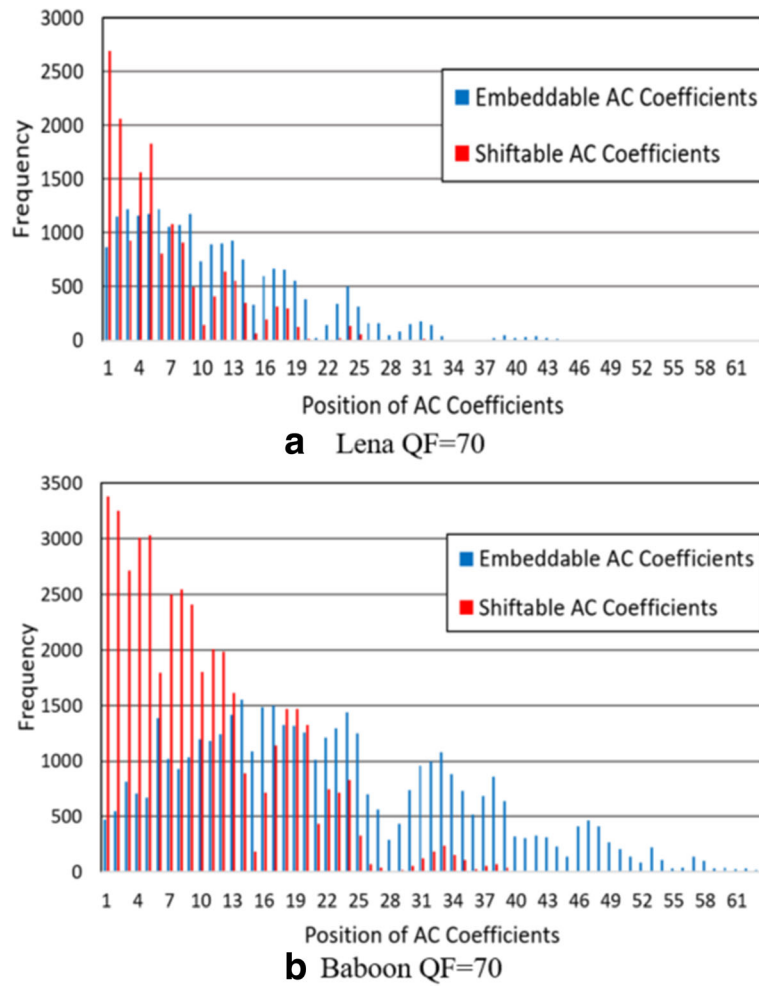


Fig. 3 Number of embeddable and shiftable AC coefficients on each position

reasons, we propose a method of selecting positions of AC coefficients for embedding.

(2) Embedding capacity and distortion: The positions are chosen by considering the embedding capacity and the distortion. The embedding capacity is measured using the number of embeddable AC coefficients. The distortion is modeled using the number of shiftable AC coefficients, quantization table, and PSNR function.

(3) PSNR function: For PSNR function, although it is not the best model for evaluating perceptual distortion, it is the most agreed upon method for measuring the distortion. It penalizes modification as a square of deviation, i.e., $f(x) = x^2$.

(4) Quantization table: The quantization table has crucial information about the compression ratio. Each entry in the quantization table defines the degree of compression (distortion) for each DCT coefficient block. This is because each DCT coefficient is divided by its corresponding quantization entry. Therefore, the modification of the quantized coefficient is directly proportional to the square of its quantization entry during the pixel reconstruction step.

Figure 4 shows the squared quantization entries for each position for $QF = 70$. It can be easily observed that modifying AC coefficients in the position 45 will cause huge distortion, while the position 1 will cause very little distortion.

(5) Embedding efficiency: In order to select the least distortion-causing positions, the proposed method has to find positions by considering the number of embeddable and shiftable AC coefficients and the quantization table entry. The proposed method calculates a metric called embedding efficiency R_i for each position $i \in 1, 2, \dots, 63$:

$$R_i = \sum_{n=1}^{4096} \frac{E_{(i,n)}}{\left(S_{(i,n)} + \frac{E_{(i,n)}}{2}\right) \times Q_i^2} \tag{4}$$

where, $E_{(i,n)} \in \{0, 1\}$ represents whether the AC coefficient in position i in the n^{th} block is embeddable (1) or not (0). Similarly, $S_{(i,n)}$ represents whether the AC coefficient in position i in the n^{th} block is shiftable (1) or not (0). Q_i is the quantization table entry at position i .

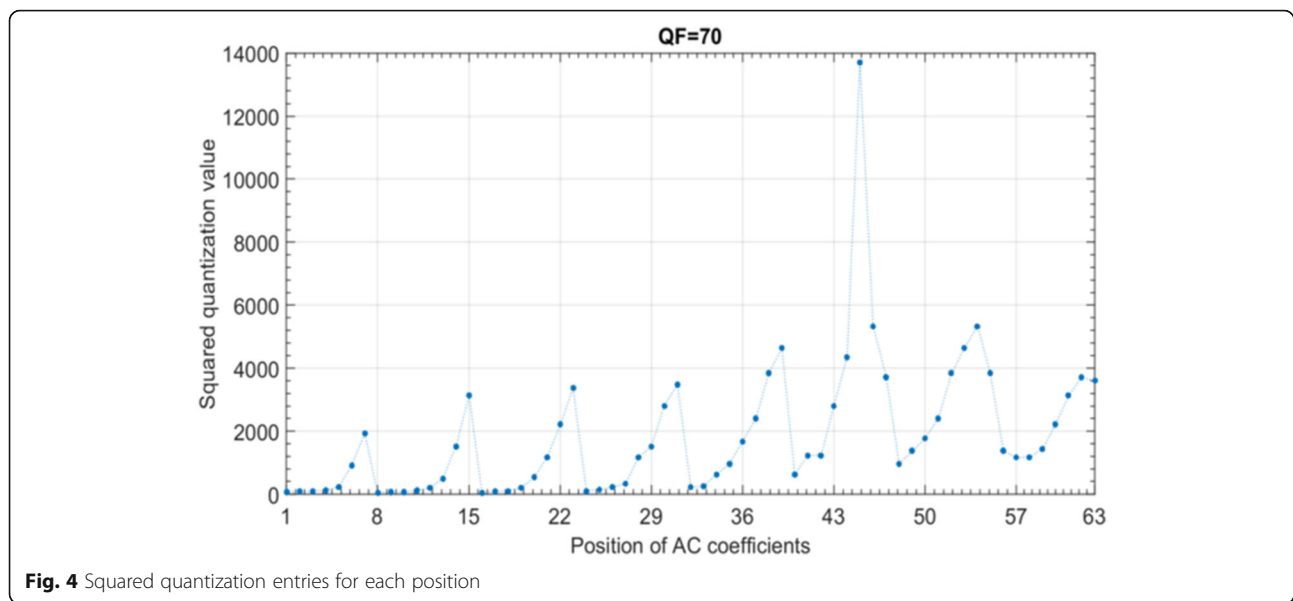


Fig. 4 Squared quantization entries for each position

Note that the nominator of the equation represents the total embedding capacity when AC coefficients in position i are used for embedding, whereas the denominator represents the corresponding estimated distortion: sum of total shiftable AC coefficients and half of the embedding capacity (assuming the payload is pseudo-random, approximately half of the embeddable will cause distortion) with the squared quantization term of penalty. The reason to squaring the quantization table entry is, to consider its effect on de-quantization phase. Although the number of embeddable coefficients are many and shiftable coefficients are small on a given position, if the quantization table entry is a large number, then image may be highly distorted after data embedding. To make clear the difference between positions with respect to the value of R , this method squared the quantization table values. So, from two positions which have similar distortion (denominator result from Eq. (4) without squaring Q_i), the one which has the smaller Q_i value will be preferred for embedding. Therefore, the best position to embed will have the largest embedding

efficiency R_i value, whereas the worst one will have the lowest value. For Eq. (4), the image with 512×512 size is considered and the number blocks becomes 4096.

(6) Sorting and position selection: Once the embedding efficiency is calculated for all 63 positions, the values can be sorted from the highest to the lowest. From here, we have to ensure that enough positions are chosen so that the payload can be embedded, this can be done by choosing the minimum number of positions from the top of the list such that the embedding capacity for each chosen positions is equal or higher than the payload. Once the positions are chosen, the embedding can take place for each block, in the order previously defined from block ordering. To make sure the encoder and the decoder uses the same order for the positions, authors suggest embedding from the lowest to the highest position, i.e., if positions (5, 1, 6, 9) are chosen from embedding, then for each block, embed in position 1 first, 5 next, then 6, and finally 9 and for extraction, extract in position 1 first, 5 next, then 6, and finally 9.

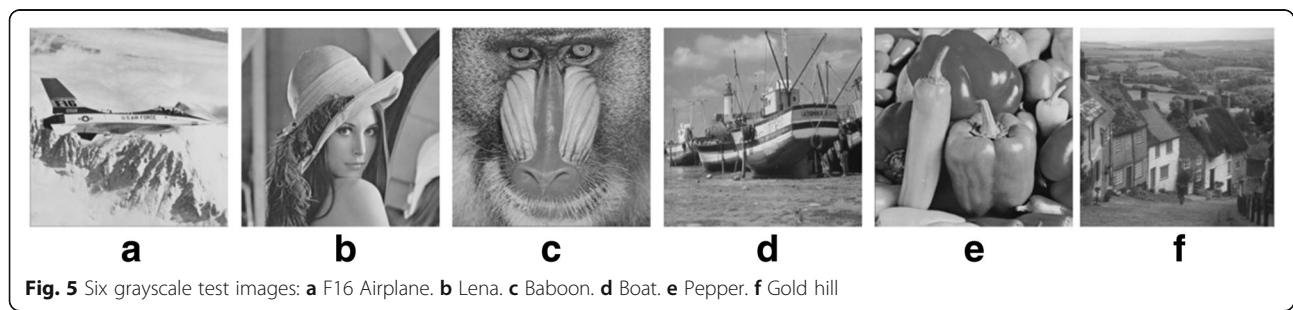


Fig. 5 Six grayscale test images: a F16 Airplane. b Lena. c Baboon. d Boat. e Pepper. f Gold hill

Table 1 Comparison in terms of PSNR (dB) with payload size of 8000 bits

Image	Methods			
	[24]	[23]	[19]	Proposed
F16	40.59	40.41	44.24	45.03
Lena	41.37	41.34	45.24	45.53
Baboon	38.03	37.95	42.49	42.83
Boat	40.41	39.74	43.43	44.40
Peppers	42.63	41.99	45.49	45.86
Gold hill	41.34	41.30	43.07	44.28
Average	40.73	40.46	43.99	44.66

Table 2 Comparison in terms of PSNR (dB) with payload size of 16,000 bits

Image	Methods			
	[24]	[23]	[19]	Proposed
F16	37.43	37.33	38.26	39.15
Lena	38.48	38.29	39.71	40.13
Baboon	34.39	34.28	36.95	37.92
Boat	37.14	36.80	38.89	39.68
Peppers	39.33	38.83	40.69	40.99
Gold hill	37.76	37.69	39.21	40.24
Average	37.42	37.20	38.95	39.69

3.4 Side information

In order to make the proposed scheme truly reversible, side information such as payload length and the positions of the embedded AC coefficients need to be embedded in the DC coefficients. The maximum payload length which must be transmitted is $\log_2(W \times H)$, where W is the width and H is the height of the image. This is 18 bits for 512×512 -sized images. The positions of AC coefficients which are used for embedding can be represented in the binary vector of size 63 bits, one bit for each position, i.e., if positions (5, 1, 6, 9) are chosen from embedding, then from the total 63, bits at vector index 1, 5, 6, and 9 will be valued 1 and the other will be valued 0. Therefore, the total side information needed for a 512×512 image is 81 bits and this can be embedded in the first 81 LSBs of the DC values. In order to

facilitate perfect recovery of the LSB of the original DC values, they are appended as part of the payload before embedding.

3.5 Extraction and image recovery

In reversible data hiding techniques, the payload should be extracted and the original image should be recovered correctly without any errors. The extraction and recovery is done simultaneously. First, the quantized DCT coefficients are retrieved. The 81 LSBs of the DC coefficients are read to find the size of the payload and the positions of the AC coefficients which were used for embedding. Then, block ordering is used in embedding order for the blocks. Since zero-valued AC coefficients are not modified, the same ordering from the embedding is achieved. Using the position of the AC coefficients which

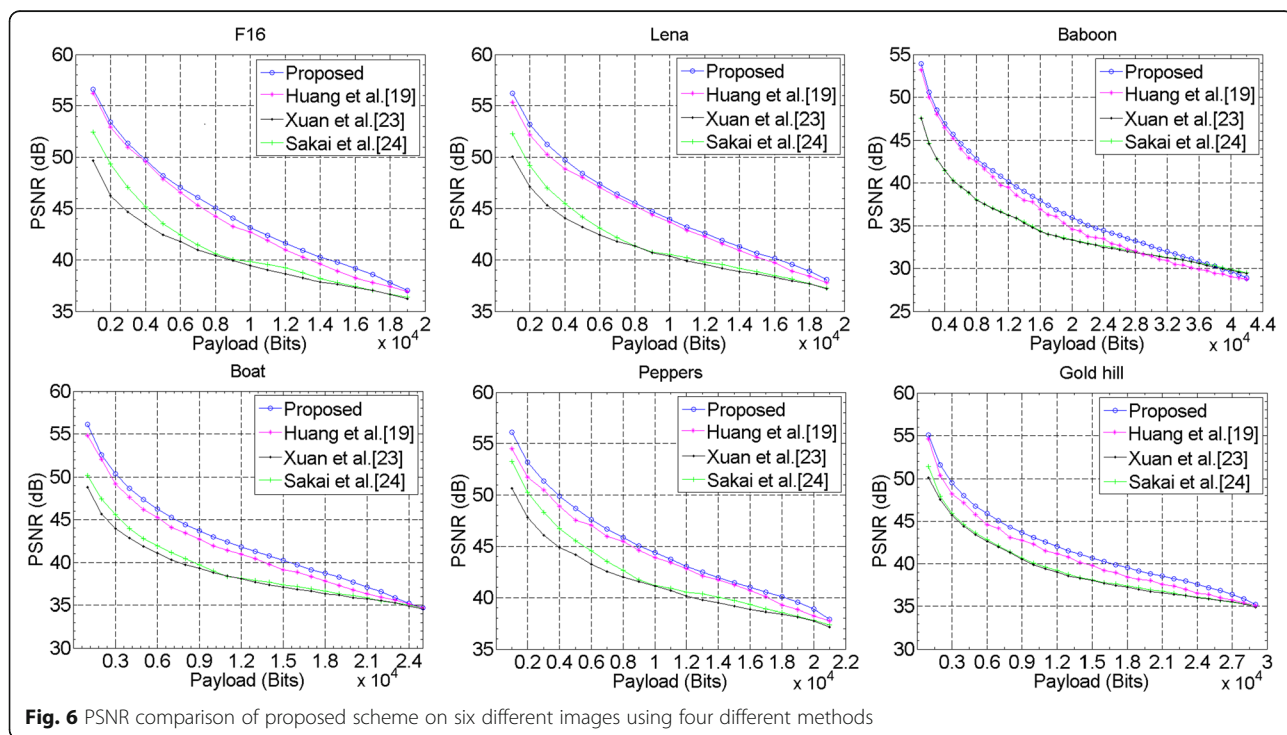


Fig. 6 PSNR comparison of proposed scheme on six different images using four different methods

were used, the original AC coefficients are recovered using Eq. (5). The payload and the original LSBs of the 81 DC coefficients are extracted using Eq. (6). Once the original LSBs of the DC coefficients are recovered, perfect recovery of the original JPEG image is completed.

$$C = \begin{cases} C' - 1, & \text{if } C' > 1 \\ C' + 1, & \text{if } C' < -1 \\ C', & \text{else} \end{cases} \quad (5)$$

$$b = \begin{cases} 1, & \text{if } |C'| = 2 \\ 0, & \text{if } |C'| = 1 \end{cases} \quad (6)$$

3.6 Complexity

The computational complexity of the proposed method is quiet minimal. To select the positions of the AC coefficients for embedding, embeddable, shiftable and unchangeable AC coefficients must be counted. This can be done quite efficiently by adding a counter loop during the Huffman code-word decoding of the quantized coefficients. For determining the efficiency ratios, multiplication (squaring the 63 entries of the quantization table) and division (63 times) are required. Finally, the sorting of 63 ratios should be not much of a task. Overall, the computational complexity is very low.

3.7 Encoder and decoder

In this section, a brief overview of the proposed encoder and decoder is presented to aid with understanding.

3.7.1 Encoder

The encoder is done as follows:

- Sort the blocks by the number of zero AC coefficients
- Count the total number of embeddable, shiftable, and unchangeable AC coefficients for each of the 63 possible positions
- Calculate the embedding efficiencies of each position and determine the positions which will lead to the lowest distortion to embed the payload (including the original LSBs of the 81 DC coefficients).
- Sort the positions using embedding efficiency R_i
- Embed the payload
- Replace the LSBs of the first DC coefficients with the side information.

3.7.2 Decoder

The decoding is done as follows:

- Read the LSBs of the first 81 DC coefficients to extract the payload size and the positions of the AC coefficients used for embedding
- Sort the blocks by the number of zero AC coefficients
- Extract the payload and recover the original AC coefficients
- Replace the LSBs of the DC coefficients with the original values (which were extracted along with the payload)

4 Experimental results

In our experiments, 24 standard bitmap images from the USC-SIPI database¹ are used. The six popular images (F16, Lena, Baboon, Boat, Pepper, and Gold hill) shown in Fig. 5a–f are used to explore the experiment in figures. All test images are 512 × 512 in size. These bitmap files are compressed using the JPEG library libjpeg (<http://www.ijg.org/>) with optimized Huffman table. The proposed method was evaluated by comparing the visual

Table 3 Comparison in terms of PSNR (dB) on USC-SIPI 23 images with payload size of 10,000 bits

Image	Methods			
	[24]	[23]	[19]	Proposed
4.2.01	42.65	42.33	43.90	44.88
4.2.02	40.85	40.64	43.35	44.13
4.2.03	37.05	36.97	40.69	41.41
4.2.04	40.51	40.22	43.69	43.94
4.2.05	39.68	39.36	42.71	43.15
4.2.06	39.23	39.08	42.82	43.23
4.2.07	41.25	41.07	43.89	44.39
5.2.08	39.38	39.21	41.47	42.72
5.2.09	37.96	37.99	40.68	41.25
5.2.10	38.64	38.52	39.90	41.01
7.1.01	41.02	40.82	42.62	43.39
7.1.02	42.15	41.31	45.96	45.99
7.1.03	41.71	41.45	42.97	43.58
7.1.04	41.89	41.58	41.97	43.23
7.1.05	40.65	40.57	40.34	41.66
7.1.06	40.69	40.68	40.30	41.66
7.1.07	41.31	41.33	40.56	41.96
7.1.08	42.64	41.96	44.54	45.01
7.1.09	40.98	40.94	41.61	42.50
7.1.10	41.85	41.67	41.92	43.07
boat.512	39.01	38.78	41.92	42.97
elaine.512	42.01	41.71	42.85	44.75
house	38.23	38.23	40.70	41.84
Average	40.49	40.28	42.23	43.12

quality and file size against the three state-of-the-art schemes [19, 23, 24]. Schemes which require modification of the quantization matrix, such as [22], are not included in the experimental result, as they usually retain higher visual quality for much larger file size. This effect can be clearly observed in [19]. The PSNR is used to evaluate the visual quality of the marked JPEG image, and it is calculated between the original JPEG image and the watermarked JPEG image. File size is measured using the number of bytes. Note that all experiments were done under $QF = 70$.

4.1 Visual quality

For visual quality comparison, the PSNR is calculated between the watermarked JPEG image and the original JPEG image. The visual quality of the cover image concern researchers to design a distortion function [28] that preserved the intrinsic statistical characteristics of the image after embedding (Tables 1 and 2).

The complete PSNR results on the six images are shown in Fig. 6. The experiment shows that the proposed method has the highest PSNR than all the previous works, but it eventually converges to [19] as payload increases. This is because AC coefficient position selection has no effect when all AC coefficients are used for embedding. Tables 1 and 2 show the numerical results of PSNR values on the listed six images, and Table 3 shows the results of the 23 images which are from the USC-SIPI database, where the data at the bottom are the average results. From the tables, the proposed technique

performs the best. The result of the proposed method seems very impressive given that the AC coefficient selection scheme requires small additional computation when compared with [19].

4.2 File Size

It is clear that file size is important to be considered with image quality in JPEG data hiding. Figure 7 shows the file size comparison. The orange dashed line in this figure is the original file size of the image. The horizontal axis represents the payload size, while the vertical axis does the image file size (in bytes). The file size of the watermarked image obtained by the proposed method is on average less than that of the previous methods (Tables 3 and 4).

Tables 5 and 6 show the numerical results of file size increment on the listed six images, and Table 4 shows results of the 23 images which are from the USC-SIPI database, where the data at the bottom are the average results. From the tables, we can see that the proposed method has the smallest file size mostly, but [24] seems to achieve the smallest file size for some images. File size is measured using the number of bytes.

5 Conclusions

The popularity and easy accessibility of the JPEG image format is becoming a great area of research for data hiding. Modification to the JPEG image introduces distortion and increase in the file size. In the proposed method, reversible data hiding for JPEG images is discussed. The

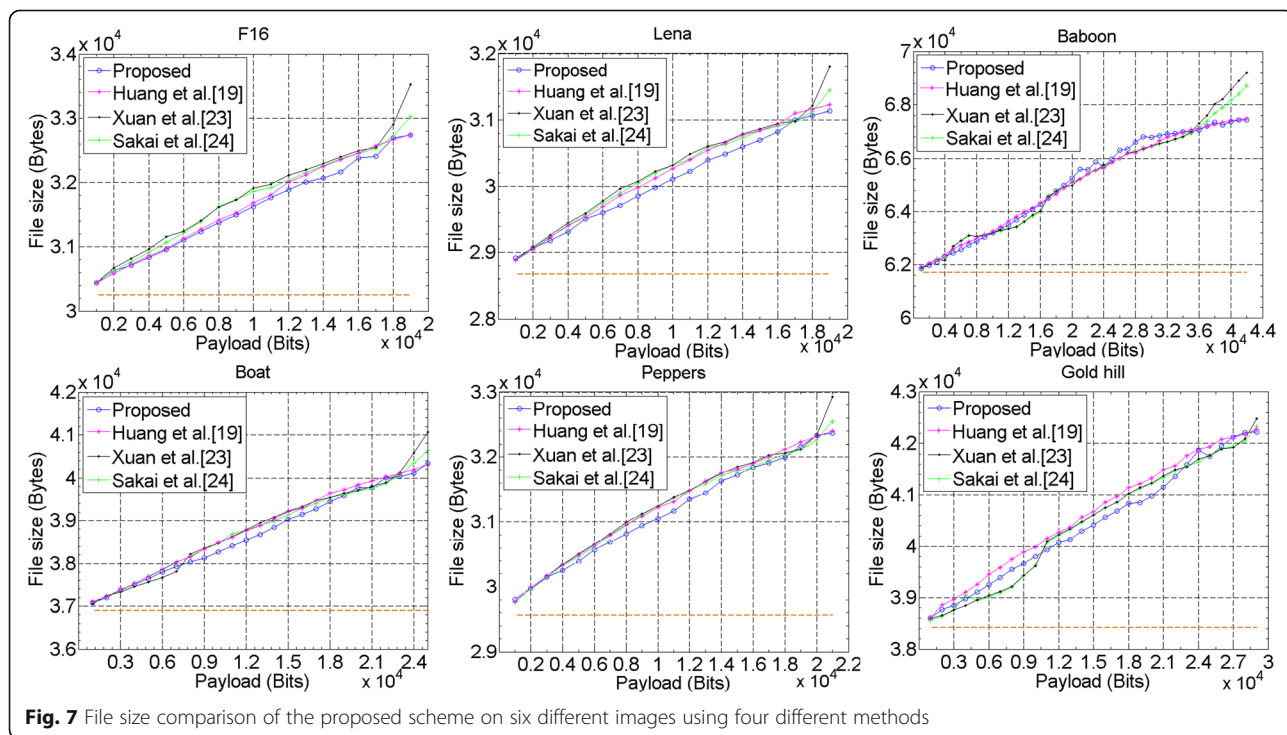


Fig. 7 File size comparison of the proposed scheme on six different images using four different methods

Table 4 Comparison in terms of increasing file size (bytes) on USC-SIPI 23 images with payload size of 10,000 bits

Image	Methods			
	[24]	[23]	[19]	Proposed
4.2.01	1674	1714	1656	1636
4.2.02	1595	1613	1556	1474
4.2.03	1501	1521	1563	1495
4.2.04	1633	1695	1587	1432
4.2.05	1574	1666	1444	1376
4.2.06	1595	1571	1449	1363
4.2.07	1672	1712	1667	1493
5.2.08	1506	1524	1495	1397
5.2.09	1233	1259	1704	1529
5.2.10	1453	1451	1712	1508
7.1.01	1290	1475	1491	1247
7.1.02	1677	1774	1515	1559
7.1.03	1392	1385	1693	1427
7.1.04	1433	1468	1755	1185
7.1.05	1171	1191	1696	1426
7.1.06	1244	1237	1720	1373
7.1.07	1179	1182	1864	1418
7.1.08	1480	1582	1495	1300
7.1.09	1264	1237	1599	1384
7.1.10	1442	1316	1737	1126
boat.512	1564	1592	1588	1364
elaine.512	1847	1844	1921	1639
house	1629	1602	1580	1459
Average	1480	1505	1630	1418

technique uses the HS strategy to embed data. From non-zero AC coefficients, only +1 and -1 are used for embedding, and the others are expanded/shifted in either direction according to their sign. Before embedding, block ordering is done and then, appropriate embeddable sections with the best coefficients for RDH are selected based on the distortion sum of AC coefficients in a section. In

Table 5 File size increase (bytes) with payload size of 8000 bits

Image	Methods			
	[24]	[23]	[19]	Proposed
F16	1377	1372	1169	1130
Lena	1375	1397	1305	1174
Baboon	1339	1375	1286	1188
Boat	1258	1315	1250	1128
Peppers	1378	1433	1400	1253
Gold hill	782	794	1329	1129
Average	1252	1281	1290	1167

Table 6 File size increase (bytes) with payload size of 16,000 bits

Image	Methods			
	[24]	[23]	[19]	Proposed
F16	2203	2255	2220	2141
Lena	2235	2274	2261	2149
Baboon	2280	2322	2633	2575
Boat	2385	2428	2391	2246
Peppers	2282	2358	2342	2282
Gold hill	2326	2327	2444	2140
Average	2285	2327	2382	2256

general, the proposed scheme has better embedding efficiency as measured by visual quality and less file size compared with previous related research. The major contribution of this paper is in exploiting Eqs. (3) and (4). Using these equations, we can choose the appropriate AC coefficients that can reduce distortion and file size.

6 Endnote

¹<http://sipi.usc.edu/database/>

Acknowledgements

Not applicable.

Funding

This work was in part supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (NRF-2015R1A2A2A0104587) and in part by the National Natural Science Foundation of China under Grant 61772572.

Availability of data and materials

Not applicable.

Authors' contributions

FW created the proposed idea, did the experiments, and drafted the manuscript. SK participated in the experiment and manuscript editing. HK and FH helped in coordinating the paper work and finishing the manuscript of the paper. All authors read and approved the final manuscript.

Authors' information

Fisseha Teju Wedaj

He received a B.Sc. degree in Information Communication Technology from Wollo University in 2010, Ethiopia, and M.Sc. degree in Information Security from Korea University, South Korea, in 2015. Currently, he is a lecturer and researcher at Adama Science and Technology University, Adama, Ethiopia. His research interests include multimedia security and machine learning. Suah Kim

He received his B.S. degree in Mathematics from the University of Waterloo, Canada. He is a PhD student of the Graduate School of Information Security, Korea University, Korea.

Hyoung Joong Kim

He is currently with the Graduate School of Information Security, Korea University, Korea. He received his B.S., M.S., and Ph.D. from Seoul National University, Korea, in 1978, 1986, and 1989, respectively. He was a professor at Kangwon National University, Korea, from 1989 to 2006. He was a visiting scholar at University of Southern California, LA, USA, from 1992 to 1993. His research interests include data hiding such as reversible watermarking and steganography.

Fangjun Huang

He is currently with the School of Data and Computer Science, Sun Yat-sen University, China. He received his B.S. degree from Nanjing University of

Science and Technology, China, in 1995, and the M.S. and Ph.D. degrees from Huazhong University of Science and Technology, China, in 2002 and 2005, respectively. From June of 2009 to June of 2010, he was a Postdoctoral Researcher at New Jersey Institute of Technology, NJ, USA. From August of 2013 to August of 2014, he was a Korea Foundation for Advanced Studies (KFAS) scholar at Korea University, Seoul, Korea. His research interests include reversible data hiding, steganography, steganalysis, and digital forensics.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

¹School of Electrical Engineering and Computing, Adama Science and Technology University, Adama, Ethiopia. ²Graduate School of Information Security, Korea University, Seoul, South Korea. ³School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. ⁴State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China.

Received: 16 February 2017 Accepted: 14 August 2017

Published online: 06 September 2017

References

- J. M. Barton, Method and Apparatus for Embedding Authentication Information Within Digital Data, U.S. Patent 5646997, 1997
- J. Tian, Reversible data embedding using a difference expansion. *IEEE Trans. Circuits Syst. Video Technol.* **13**(8), 890–896 (2003)
- H.J. Kim, V. Sachnev, Y.Q. Shi, J. Nam, H.G. Choo, A novel difference expansion transform for reversible data embedding. *IEEE Trans. Inf. Forensics Secur.* **3**(3), 456–465 (2008)
- V. Sachnev, H.J. Kim, J. Nam, S. Suresh, Y.Q. Shi, Reversible watermarking algorithm using sorting and prediction. *IEEE Trans. Circuits Syst. Video Technol.* **19**(7), 989–999 (2009)
- Z. Ni, Y.Q. Shi, N. Ansari, W. Su, Reversible data hiding. *IEEE Trans. Circuits Syst. Video Technol.* **16**(3), 354–362 (2006)
- X. Qu, S. Kim, H.J. Kim, Reversible watermarking based on compensation. *J. Electrical Eng. Technol.* **9**, 422–428 (2015)
- X. Qu, H.J. Kim, Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding. *Signal Process.* **111**, 249–260 (2015)
- X. Li, J. Li, B. Li, and B. Yang, High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion, *Signal Process.*, vol. 93, no. 1, pp. 198–205, 2013
- F. Peng, X. Li, B. Yang, Improved pvo-based reversible data hiding. *Digit. Signal Process.* **25**, 255–265 (2014)
- I.C. Dragoi, D. Coltuc, Local-prediction-based difference expansion reversible watermarking. *IEEE Trans. Image Process.* **23**(4), 1779–1790 (2014)
- B.Y. Lee, H.J. Hwang, H.J. Kim, Reversible data hiding using piecewise autoregressive predictor based on two-stage embedding. *J. Electrical Engin. Technol.* **11**(4), 974–986 (2016)
- H.J. Hwang, S.H. Kim, H.J. Kim, Reversible data hiding using least square predictor via LASSO. *EURASIP J Image Video Process* **42**, 2016 (2016)
- W. B. Pennebaker, and J. L. Mitchell, *JPEG: Still Image Data Compression Standard*, Springer, 1993
- G.K. Wallace, The JPEG still picture compression standard. *IEEE Trans. Consumer Electron.* **38**(1), 30–44 (1992)
- Z. Qian, X. Zhang, Lossless data hiding in JPEG bit-stream. *J. Syst. Softw.* **85**, 309–313 (2012)
- B.G. Mobasser, R.J. Berger, M.P. Marcinak, Y.J. Naik Raikar, Data embedding in JPEG bit-stream by code mapping. *IEEE Trans. Image Process.* **19**(4), 958–966 (2010)
- Y. Hu, K. Wang, Z.-M. Lu, An improved VLC-based lossless data hiding scheme for JPEG images. *J. Syst. Softw.* **86**, 2166–2173 (2013)
- C.-C. Chang, C.-C. Lin, C.-S. Tseng, W.-L. Tai, Reversible hiding in DCT-based compressed images. *Inf. Sci.* **177**(13), 2768–2786 (2007)
- F. Huang, X. Qu, H.J. Kim, Reversible data hiding in JPEG image. *IEEE Trans. Circuits Syst. Video Technol.* **26**(9), 1610–1621 (2016)
- J. Fridrich and M. Goljan, Lossless data embedding for all image formats, in: *SPIE Proceedings of Photonics West, Electronic Imaging, Security and Watermarking of Multimedia Contents*, vol. 4675, pp. 572–583, 2002
- J. Fridrich, M. Goljan, and R. Du, Invertible authentication watermark for JPEG images, *IEEE International Conference on Information Technology: Coding and Computing*, pp. 223–227, 2001
- K. Wang, Z.-M. Lu, Y.-J. Hu, A high capacity lossless data hiding scheme for JPEG images. *J. Syst. Softw.* **86**, 1965–1975 (2013)
- G. Xuan, Y. Q. Shi, Z. Ni, P. Chai, X. Cui and X. Tong, Reversible data hiding for JPEG images based on histogram pairs, In *Proceedings of the International Conference on Image Analysis and Recognition*, Montreal, Canada, pp. 715–727. 2007
- H. Sakai, M. Kuribayashi, and M. Morii, Adaptive reversible data hiding for JPEG images, in *Proceedings of the International Symposium on Information Theory and Its Applications*, Auckland, New Zealand, pp. 1–6, 2008
- Q. Li, Y. Wu, F. Bao, A reversible data hiding scheme for JPEG images. *Lect. Notes Comput. Sci.* **6297**, 653–664 (2010)
- Y.K. Lin, High capacity reversible data hiding scheme based upon discrete cosine transformation. *J. Syst. Softw.* **85**(10), 2395–2404 (2012)
- M.U. Celik, G. Sharma, A.M. Tekalp, E. Saber, Lossless generalized-LSB data embedding. *IEEE Trans. Image Process.* **14**(2), 253–266 (2005)
- F. Huang, W. Luo, J. Huang, and Y. Q. Shi, Distortion function designing for JPEG steganography with uncompressed side-image. *Proceedings of the first ACM Workshop on Information hiding and Multimedia Security*, New York, USA, pp. 69–76, 2013

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com