Signal Processing 148 (2018) 41–47

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Reversible data hiding in JPEG image based on DCT frequency and block selection☆

Dongdong Hou, Haoqian Wang, Weiming Zhang*, Nenghai Yu

School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China

ARTICLE INFO

Article history:
Received 30 July 2017
Revised 10 November 2017
Accepted 1 February 2018
Available online 7 February 2018

Keywords:
JPEG image
DCT block
Frequency selection
Block selection
Reversible data hiding

ABSTRACT

The joint photographic experts group (JPEG) is the most popular format of digital image. In this paper, a novel reversible data hiding scheme based on JPEG image is proposed. In JPEG image, quantified DCT (discrete cosine transform) coefficients with different frequencies will yield different capacities and embedding distortions. To reduce the total distortion for the marked image, we select coefficients from frequencies yielding less distortions for embedding, and then an advanced block selection strategy is applied to always modify the block yielding less simulated distortion firstly until the given payloads are completely embedded. Experimental results show that the proposed method can keep good visual quality with small bitstream expansion.

1. Introduction

Reversible data hiding (RDH) [1] is one special type of information hiding, by which the host sequence as well as the embedded data can be both restored from the marked sequence without loss. This important technique is widely used in medical imagery, military imagery and law forensics, where the original signal is so precious that it cannot be damaged.

Many RDH algorithms have been proposed in the past decade, which can be roughly classified into three fundamental strategies: lossless compression appending scheme [2], difference expansion [3] and histogram shift (HS) [4]. The advanced RDH techniques combine these strategies to prediction errors (PEs) [5–10] to achieve better performances. RDH algorithms in uncompressed images have been well established. Compared with uncompressed images compressed images have less redundancy, thus are considerably more difficult for RDH. Among all the formats of compressed images, joint photographic experts group (JPEG) is the most popular one, thus RDH in JPEG image is important and useful for many applications.

As for JPEG image, except for the visual quality of marked image the file size of marked image must be also considered, because invalid modification may lead to the serious problem of bitstream expansion. That is to say the file size of generated marked JPEG image will increase seriously comparing to that of host JPEG image. Although some works [11–13] embed messages by modifying Huffman table, which can well preserve the file size of JPEG image, their embedding capacities are rather limited.

The most popular approaches [14–21] for RDH in JPEG image are based on modifying quantified DCT (discrete cosine transform) coefficients. Note that, by Wang et al.'s method [21], not only quantified DCT coefficients but also quantification table are modified to achieve high capacity and high image quality, but their method will also cause relatively large bitstream expansion. What is more, Wang et al.'s method can not be applied to JPEG image with $QF = 100$ (quality factor), the reason is that the corresponding quantification table can not be modified by Wang et al.'s method. Recently, Huang et al. [22] propose a new HS-based RDH scheme for JPEG image, by which zero coefficients remain unchanged and only coefficients with values "1" and "−1" are selected to carry messages. Moreover, a block selection strategy based on the number of zero coefficients in each 8×8 block is presented to adaptively choose DCT blocks for embedding, so that achieving high embedding capacity and well preserving the visual quality and the file size of marked JPEG image. However, one DCT block having more zero coefficients does not always means current DCT block will yield smaller distortion after embedding, and quantification table is also not considered in Huang et al.'s algorithm.

The quantified DCT coefficients with different frequencies will cause different embedding distortions and provide different capacities, i.e., the number of "1" and "−1" in alternating current (AC) coefficients. In this paper, based on HS-based embedding method

☆ This work was supported in part by the Natural Science Foundation of China under Grant 61572452, U1636201.

* Corresponding author.
E-mail addresses: houdd@mail.ustc.edu.cn (D. Hou), 17114010013@fudan.edu.cn (H. Wang), zhangwm@ustc.edu.cn (W. Zhang), ynh@ustc.edu.cn (N. Yu).

https://doi.org/10.1016/j.sigpro.2018.02.002
0165-1684/

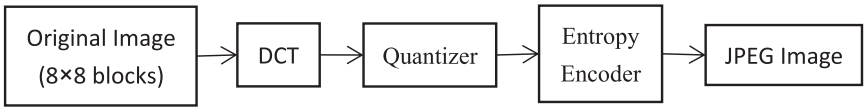


Fig. 1. Block diagram of JPEG compression processes.

on "1" and "−1" of AC coefficients, we estimate the unit distortion for each frequency, and select quantified coefficients at the first K frequencies with the minimum unit distortions for embedding. An advanced block selection strategy is also applied, by which the corresponding embedding distortion of each DCT block is simulated, and we always select the block with less simulated distortion for embedding firstly until the given payloads are completely embedded. Experimental results show that, the proposed method outperforms the previous works, thus is a more advanced RDH scheme in JPEG image.

The rest of the paper is organized as follows. We briefly introduce the previous arts in Section 2. Section 3 elaborates the proposed RDH method. Experimental results are given in Section 4 to show the advantages of the proposed method, and finally this paper is concluded with a discussion in Section 5.

2. Previous arts

Throughout this paper, matrices and vectors are denoted by boldface, and $\mathbf{X} = \{x_i\}$ means that x_i is the symbol of element in matrix $\mathbf{X}$.

The JPEG encoder mainly consists of three components, that is DCT, quantizer, and entropy encoder. By applying two-dimensional DCT to each non-overlapping 8×8 block, original image data is transformed from spatial domain to frequency domain. These obtained DCT coefficients are then quantified according to quantification table. The quantified DCT coefficients are arranged in a zigzag scanning order and pre-compressed using the differential pulse code modulation on direct current coefficients and run length encoding on AC coefficients. Finally, the symbol string is Huffman coded to obtain the final compressed bitstream. After pre-pending the header, we obtain the final JPEG image. The main steps of JPEG compression are depicted as Fig. 1.

Mathematical definitions of 8×8 forward DCT and 8×8 inverse DCT (IDCT) are formulated as Eq. (1) and Eq. (2) respectively.

$$F(u,v) = \frac{1}{4}c(u)c(v)\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16} \tag{1}$$

$$f(x,y) = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} c(u)c(v)F(u,v)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16} \tag{2}$$

where:

$$c(u) = \begin{cases} \frac{1}{\sqrt{2}}, & if\ u = 0 \\ 1, & otherwise \end{cases}. \tag{3}$$

To compress image data, these DCT coefficients are quantified by using a quantification table with 64 entries, and a standard JPEG quantification table is shown as Fig. 2. The quantified coefficients are all integers which are obtained through dividing each DCT coefficient by its corresponding step in quantification table and rounding it to the nearest integer as follows:

$$d(u,v) = IntegerRound\left(\frac{F(u,v)}{q(u,v)}\right), \tag{4}$$

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

Fig. 2. Standard JPEG quantification table.

where $F(u, v)$ is the original DCT coefficient with the frequency $\{u, v\}$, $q(u, v)$ is the corresponding step in quantification table $\mathbf{Q}$, i.e., at the position of the uth row and the vth col of block, and $d(u, v)$ is the quantified DCT coefficient in block $\mathbf{D}$. Most of works as well as the presented work embeds messages into quantified DCT coefficients.

Without any loss of generality, the nonzero AC coefficients are collected. According to Huang et al.'s [22] analysis, the peak points of nonzero AC coefficient histogram are generally located at points "1" and "−1". Bin 1 and −1 are grouped into the inner region to carry messages, and the rest of bins are grouped into the outer region. The embedding manner is described as follows:

$$d'(u,v) = \begin{cases} d(u,v) + sign(d(u,v)) * b, & if\ |d(u,v)| = 1 \\ d(u,v) + sign(d(u,v)), & if\ |d(u,v)| > 1 \end{cases}, \tag{5}$$

where

$$sign(x) = \begin{cases} 1, & if\ x > 0 \\ 0, & if\ x = 0 \\ -1, & if\ x < 0 \end{cases}. \tag{6}$$

In Eq. (5), $b \in \{0, 1\}$ denotes one bit message to be embedded and $d'(u, v)$ represents the AC coefficient after embedding. As shown in Eq. (5), all coefficients are modified at most by one in the embedding processes. Obviously, the message extraction and image restoration can be achieved by Eqs. (7) and (8).

$$b = \begin{cases} 0, & if\ |d'(u,v)| = 1 \\ 1, & if\ |d'(u,v)| = 2 \end{cases} \tag{7}$$

$$d(u,v) = \begin{cases} sign(d'(u,v)), & if\ 1 \le |d'(u,v)| \le 2 \\ d(u,v) - sign(d'(u,v)), & if\ |d'(u,v)| > 2 \end{cases} \tag{8}$$

Huang et al. embed messages block by block, to reduce the distortion for marked image, they preferentially select blocks with more zero coefficients for embedding. Such block selection strategy reduces embedding distortion a lot.

3. Proposed scheme

3.1. Select frequencies to extract host coefficients

As for the block-by-block embedding strategy, assume the capacity for the ith DCT block is C_i, and the corresponding embedding distortion is J_i, where $1 \le i \le N$ and N is the number of DCT blocks. The embedding distortion J_i and capacity C_i are determined by the embedding strategy, and a better embedding strategy will introduce the less J_i, or provide the larger C_i. For one DCT block all the AC coefficients with values "1" and "−1" are selected to carry messages and the rest of nonzero AC coefficients apart from values "1" and "−1" will be shifted by Huang et al.'s method [22]. Therefore, C_i is the number of AC coefficients with values "1" and "−1" in the ith DCT block and J_i is caused by modifying all of its nonzero AC coefficients. However, sometimes such embedding method will add unnecessary distortion. For example, if there is nearly no value "1" or "−1" at one frequency, but many other nonzero AC values exist. There is no doubt that AC coefficients with such frequency should be skipped for modification, then we can reduce J_i while keeping C_i. On the other hand, modifying coefficients with different frequencies will yield different distortions, and which frequencies deserve the priority for modification is also not considered deeply in the previous works. The above faultiness inspires us that, we can reduce the total embedding distortion by properly selecting frequencies for modification.

In the processes of IDCT, we firstly recover original DCT coefficients with

$$F(u,v) = d(u,v) \times q(u,v), \tag{9}$$

and then return these recovered coefficients to pixel values with Eq. (2). The inverse transformation between 8×8 quantified DCT block $\mathbf{D} = \{d(u,v)\}$ and 8×8 spatial pixel block $\mathbf{P} = \{f(x,y)\}$ is simplified as

$$\mathbf{P} = DCT(\mathbf{D} \odot \mathbf{Q}), \tag{10}$$

where $\mathbf{D} \odot \mathbf{Q} = \{d(u,v) * q(u,v)\}$ means multiplying each value in $\mathbf{D}$ by value at the same position in $\mathbf{Q}$.

Indeed, adding a number to the quantified DCT coefficient $d(u, v)$ is equivalent to adding the number multiplied by the corresponding quantification step $q(u, v)$ to original DCT coefficient $F(u, v)$. Therefore, the modifications on DCT coefficients at frequencies with large quantification steps will yield larger distortions.

If we add 1 to $d(u, v)$ and keep the other coefficients unchanged, then the yielded average distortion for 8×8 DCT block denoted as $cost(u, v)$ is

$$cost(u,v) = \frac{\sum_{x=0}^{7}\sum_{y=0}^{7}\Delta f(x,y)^2}{64}, \tag{11}$$

where

$$\begin{aligned}\Delta f(x,y) &= f'(x,y) - f(x,y)\\ &= \frac{1}{4}\sum_{i=0}^{7}\sum_{j=0}^{7} c(i)c(j)\Delta F(i,j)\cos\frac{(2x+1)i\pi}{16}\cos\frac{(2y+1)j\pi}{16}\\ &= \frac{1}{4}c(u)c(v)q(u,v)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}.\end{aligned} \tag{12}$$

The quantified DCT coefficient at frequency $\{u, v\}$ of the kth DCT block is denoted as $d^k_{u,v}$. We collect quantified DCT coefficients at frequency $\{u, v\}$ from all the DCT blocks as a vector denoted by $\mathbf{D}_{u,v}$, such that $\mathbf{D}_{u,v} = \{d^1_{u,v}, d^2_{u,v}, \ldots, d^N_{u,v}\}$, where N is the number of DCT blocks. To evaluate the embedding performance of DCT coefficients at the frequency $\{u, v\}$, we calculate its unit distortion for embedding per bit. As utilized in Huang et al.'s work [22], we select AC coefficients with values "1" and "-1" for carrying messages and shift the rest of nonzero AC coefficients. Therefore, the embedding capacity for vector $\mathbf{D}_{u,v}$ denoted as $C_{u,v}$ is the number of "1" and "−1" in $\mathbf{D}_{u,v}$, and the total distortion caused by embedding $C_{u,v}$ bits messages into $\mathbf{D}_{u,v}$ is estimated as

$$J_{u,v} = (0.5 * C_{u,v} + C_{out}) * cost(u,v), \tag{13}$$

where C_{out} is the number of nonzero AC coefficients apart from values "1" and "−1".

Now, as for $\mathbf{D}_{u,v}$, we can exactly calculate its capacity $C_{u,v}$ and roughly estimate corresponding embedding distortion $J_{u,v}$, thus unit distortion for frequency $\{u, v\}$ (denoted as $UD_{u,v}$)is

$$UD_{u,v} = J(u,v)/C_{u,v}. \tag{14}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{15}$$

We sort 63 frequencies of AC coefficients according to their unit distortions, and select the first K frequencies with the minimum unit distortions to extract host coefficients. There are at most 63 choices for K, after the payloads are given for host image we can easily get a suitable K by exhaustive search. After K frequencies are decided, we extract a sub-block denoted by $\mathbf{D}'$ from $\mathbf{D}$, such that

$$\mathbf{D}' = \mathbf{B} \odot \mathbf{D}, \tag{16}$$

where $\mathbf{B}$ is a 1/0 matrix declaring which frequencies are selected or not. For example, if frequencies {1, 2}, {2, 1} and {2, 2} are selected for embedding, then $\mathbf{B}$ becomes Eq. (15).

3.2. Select DCT blocks according to simulated distortions

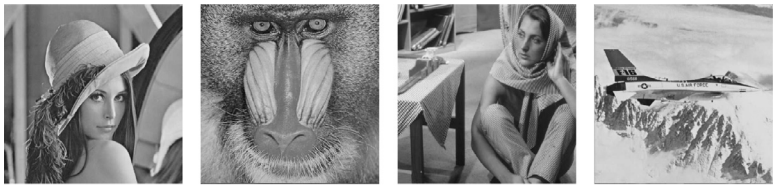
After frequencies are selected, we extract a sub-block $\mathbf{D}'$ from each quantified DCT block $\mathbf{D}$. There is no doubt that to reduce the total embedding distortion the selected host coefficients should contain shifted values in the outer region as few as possible. More zero AC coefficients in one DCT block usually means that current block belongs to the smoother area of image, and the proportion of values "1" and "−1" is high. Huang et al. select blocks with more zero coefficients for embedding, and thus reducing embedding distortion a lot. However, one DCT block having more zero coefficients does not always means current DCT block will yield less embedding distortion, because modifications on coefficients with different quantification steps will yield different embedding distortions.

Base on the above analysis, to reduce the embedding distortion, we calculate simulated distortion (SD) for each sub-block, and these sub-blocks with small SDs are preferred for modification. The smaller SD for one block usually means that not only the number of zero AC coefficients is larger, but also the nonzero AC coefficients are at frequencies with smaller quantification steps. In order to regain SD for each sub-block exactly at the receiver's side, we assume that all the embedded bits are with the value 1, and simulate modification processes as

$$d''^k(u,v) = d'^k(u,v) + sign(d'^k(u,v)), \quad if\ |D^k_{u,v}| \ge 1, \tag{17}$$

then $\mathbf{D}'^k = \{d'^k(u,v)\}$ becomes $\mathbf{D}''^k = \{d''^k(u,v)\}$. With Eq. (10) we can retransform the differences in frequency domain denoted as $\Delta\mathbf{D}'^k = \{sign(d'^k(u,v)\}$ to the differences in spatial domain denoted as $\Delta\mathbf{P}'^k = \{\Delta f'^k(u,v))\}$. Then, the simulated distortion is

$$SD_k = \sum_{x=0}^{7}\sum_{y=0}^{7}\Delta f'^k(x,y)^2. \tag{18}$$



(a) (b) (c) (d)

Fig. 3. Tested images of size 512 × 512. (a) Lena. (b) Baboon. (c) Barbara. (d) Airplane.

Table 1
Comparisons of three methods on test images in Fig. 3.

Lena QF=70	Payloads (bits)	6000	8000	10000	12000	14000
	Wang et al.'s method	(53.51,11500)	(53.42,11500)	(53.58,17648)	(52.56,17536)	(52.82,23608)
	Huang et al.'s method	(47.12,8080)	(45.21,10240)	(43.60,12536)	(42.24,14736))	(41.00,16504))
	The proposed method	(47.34,8160)	(46.58,10008)	(44.50,12120))	(42.70,14248))	(41.01,16184)
Lena, QF=80	Payloads (bits)	6000	8000	10000	12000	14000
	Wang et al.'s method	(55.66,11968)	(55.70,12120)	(54.53,18360)	(54.48,18504)	(53.60,24600)
	Huang et al.'s method	(49.84,8336)	(48.33,10312)	(46.92,12424)	(45.83,14440)	(44.50,16632)
	The proposed method	(50.23,8080)	(48.64,9960)	(47.32,12024)	(46.10,13832)	(44.88,16560)
Lena, QF=90	Payloads (bits)	6000	10000	14000	18000	22000
	Wang et al.'s method	(58.67,16048)	(57.88,22728)	(56.63,30840)	(55.92,39368)	(55.34,47848)
	Huang et al.'s method	(53.38, 9448)	(50.72,13824)	(48.79,18504)	(47.25,22440)	(45.79,27272)
	The proposed method	(54.14, 8680)	(51.31,13040)	(49.19,17184)	(47.50,21432)	(46.16,25872)
Lena, QF=100	Payloads (bits)	8000	16000	24000	32000	40000
	Wang et al.'s method	–	–	–	–	–
	Huang et al.'s method	(58.43,9960)	(54.80,23168)	(52.40,29784)	(50.95,38272)	(49.45,50984)
	The proposed method	(59.93, 7712)	(56.39,17032)	(54.32,25920)	(52.74,36320)	(51.44,43880)
Baboon, QF=70	Payloads (bits)	6000	8000	10000	12000	14000
	Wang et al.'s method	(53.60,16288)	(53.92,16208)	(52.79,22592)	(53.12,22976)	(51.72,29056)
	Huang et al.'s method	(44.26,7576)	(42.30,9888)	(40.81,12568)	(39.61,15184)	(38.16,17864)
	The proposed method	(44.70,7320)	(42.75,9408)	(41.21,11656)	(39.67,14264)	(38.35,16992)
Baboon, QF=80	Payloads (bits)	6000	8000	10000	12000	14000
	Wang et al.'s method	(55.18,16736)	(55.68,16856)	(54.40,24288)	(54.87,24400)	(53.49,30224)
	Huang et al.'s method	(45.95, 7368)	(44.18, 9992)	(42.61,12152)	(41.42,15352)	(40.04,18016)
	The proposed method	(46.30, 6936)	(44.41, 9440)	(42.76,11944)	(41.41,14520)	(40.20,16880)
Baboon, QF=90	Payloads (bits)	6000	10000	14000	18000	22000
	Wang et al.'s method	(58.59,19208)	(57.86,26840)	(56.59,37936)	(55.98,46608)	(55.35,55696)
	Huang et al.'s method	(48.29, 8152)	(45.22,13768)	(43.18,19416)	(41.38,24672)	(39.91,30696)
	The proposed method	(48.49, 8192)	(45.41,13448)	(43.15,19160)	(41.39,24552)	(39.98,29240)
Baboon, QF=100	Payloads (bits)	8000	16000	24000	32000	40000
	Wang et al.'s method	–	–	–	–	–
	Huang et al.'s method	(59.61, 9056)	(56.26,16144)	(54.22,21480)	(52.58,28360)	(51.22,35208)
	The proposed method	(60.40, 7784)	(57.27,15896)	(55.32,23920)	(53.88,30456)	(52.52,35768)
Barbara, QF=70	Payloads (bits)	6000	8000	10000	12000	14000
	Wang et al.'s method	(53,98,12400)	(54.26,12240)	(52.99,18680)	(52.58,18824)	(52.01,25864)
	Huang et al.'s method	(44.40,7032)	(42.57,9664)	(40.86,12416)	(39.85,14656)	(38.77,16776)
	The proposed method	(45.62,6888)	(43.36,6888)	(41.65,12160)	(40.42,14432)	(39.18,16880)
Barbara, QF=80	Payloads (bits)	6000	8000	10000	12000	14000
	Wang et al.'s method	(55.37,13048)	(55.78,13072)	(54.66,19400)	(55.07,20024)	(53.62,26144)
	Huang et al.'s method	(48.44, 7312)	(45.93, 9896)	(44.20,12304)	(42.60,14760)	(41.45,17208)
	The proposed method	(49.34, 7088)	(46.66,10144)	(44.74,12448)	(43.31,14832)	(42.00,17352)
Barbara, QF=90	Payloads (bits)	6000	10000	14000	18000	22000
	Wang et al.'s method	(58.65,14960)	(57.82,20408)	(56.69,29800)	(55.99,38192)	(55.43,46288)
	Huang et al.'s method	(54.69, 6552)	(51.39,11328)	(47.79,17040)	(45.34,21688)	(43.45,26400)
	The proposed method	(55.08, 6568)	(51.67,10792)	(48.15,16464)	(45.48,21568)	(43.57,25616)
Barbara, QF=100	Payloads (bits)	8000	16000	24000	32000	40000
	Wang et al.'s method	–	–	–	–	–
	Huang et al.'s method	(59.88, 8600)	(56.59,15216)	(54.53,20280)	(52.88,26744)	(51.47,33784)
	The proposed method	(60.27, 8040)	(57.27,15712)	(55.30,24176)	(53.77,30904)	(52.48,36288)
Airplane, QF=70	Payloads (bits)	6000	8000	10000	12000	14000
	Wang et al.'s method	(54.02,11928)	(54.36,11952)	(53.07,17984)	(53.65,17864)	(52.00,24744)
	Huang et al.'s method	(46.87,6944)	(44.41,9360)	(43.09,11752)	(41.30,13856)	(39.86,15920)
	The proposed method	(47.79,6680)	(45.72,9008)	(43.87,10912)	(42.02,13792)	(40.45,15782)
Airplane, QF=80	Payloads (bits)	6000	8000	10000	12000	14000
	Wang et al.'s method	(55.32,12440)	(55.71,12200)	(54.62,18144)	(55.06,18232)	(53.64,24576)
	Huang et al.'s method	(49.77, 6912)	(48.11, 9024)	(46.66,10952)	(45.32,13072)	(43.72,15528)
	The proposed method	(50.70, 6544)	(48.90, 8912)	(47.35,10856)	(45.85,12912)	(44.31,15592)
Airplane, QF=90	Payloads (bits)	6000	10000	14000	18000	22000
	Wang et al.'s method	(58.79,15448)	(57.91,21096)	(56.70,30264)	(55.98,38856)	(55.38,47240)
	Huang et al.'s method	(54.10, 8648)	(51.22,13264)	(49.31,17040)	(47.25,22328)	(45.16,25648)
	The proposed method	(54.70, 8168)	(51.91,12328)	(49.80,16664)	(47.32,21496)	(45.11,25304)
Airplane, QF=100	Payloads (bits)	8000	16000	24000	32000	40000
	Wang et al.'s method	–	–	–	–	–
	Huang et al.'s method	(62.30, 6208)	(58.71,13184)	(56.64,21720)	(54.92,29608)	(53.57,40248)
	The proposed method	(62.45, 5744)	(58.95,12656)	(56.85,18840)	(55.24,27144)	(53.89,36312)

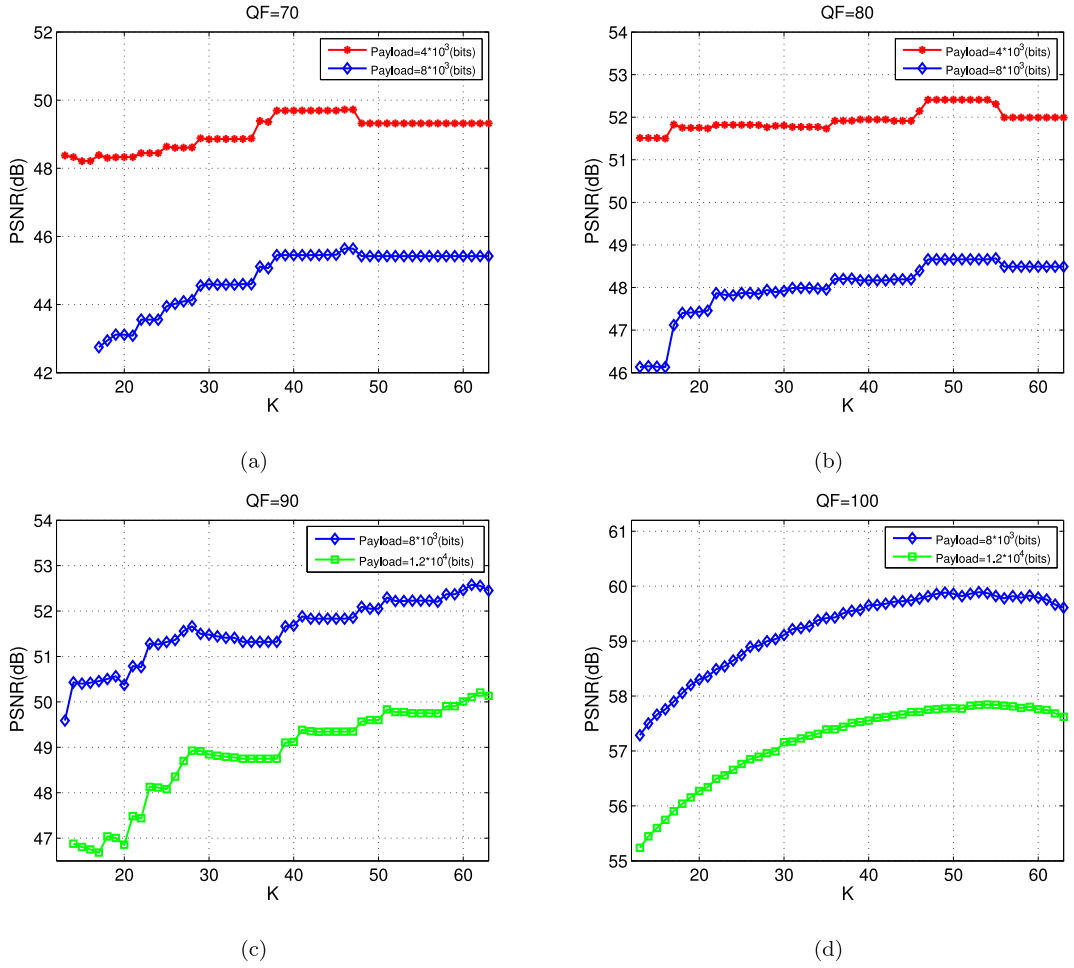


Fig. 4. PSNR values of marked images under two payloads by setting different K values.

The HS-based modification (Eq. (5)) on each coefficient will not change its sign (positive or negative), thus Eq. (19) holds.

$$\Delta d'^{k}(u,v) = sign(d''^{k}(u,v)) = sign(d'^{k}(u,v)) \tag{19}$$

At the receiver's side, according to the marked coefficients we can recalculate these differences in frequency domain with Eq. (19), then regain SD for each sub-block and relocate those sub-marked-blocks carrying messages.

We sort these sub-blocks according to their simulated distortions, and always select the sub-block with the smaller SD for embedding firstly until the selected sub-blocks can accommodate the given payloads.

3.3. Data embedding and extracting

To restore host image and embedded messages, we need to record the volume of embedded messages and the selected frequencies, which are regarded as a part of the payloads to be embedded. Now, the detailed processes of embedding and extracting are described as follows:

3.3.1. Embedding

1. Decode original JPEG image to get the quantified DCT coefficients, and then compute unit distortion for each frequency;
2. Extract sub-blocks according to selected K frequencies, and these frequency information are regarded as a part of the payloads to be embedded;
3. Calculate simulated distortion for each sub-host-block, and embed messages into sub-host-blocks yielding less simulated distortions firstly until the given payloads are completely embedded;
4. Return all the modified coefficients to their original positions and entropy-encode the obtained coefficients to get the marked JPEG image.

3.3.2. Extracting

1. Decode marked JPEG image to get the quantified DCT coefficients;
2. Extract sub-blocks according to the extracted frequency information;
3. Simulate embedding distortion for each sub-block, and extract messages from sub-marked-blocks yielding less simulated distortions firstly until the embedded messages are completely extracted;
4. Encode the restored coefficients again to get the host JPEG image.

4. Experimental results

When host signal for RDH is JPEG image, invalid modification may lead to serious problem of bitstream expansion, i.e., the increased file size between host image and marked image. Therefore, for each experiment, not only Power Signal-to-Noise Ratio (PSNR) but also bitstream expansion is adopted to evaluate the performance of an RDH scheme in JPEG image. All the test images are

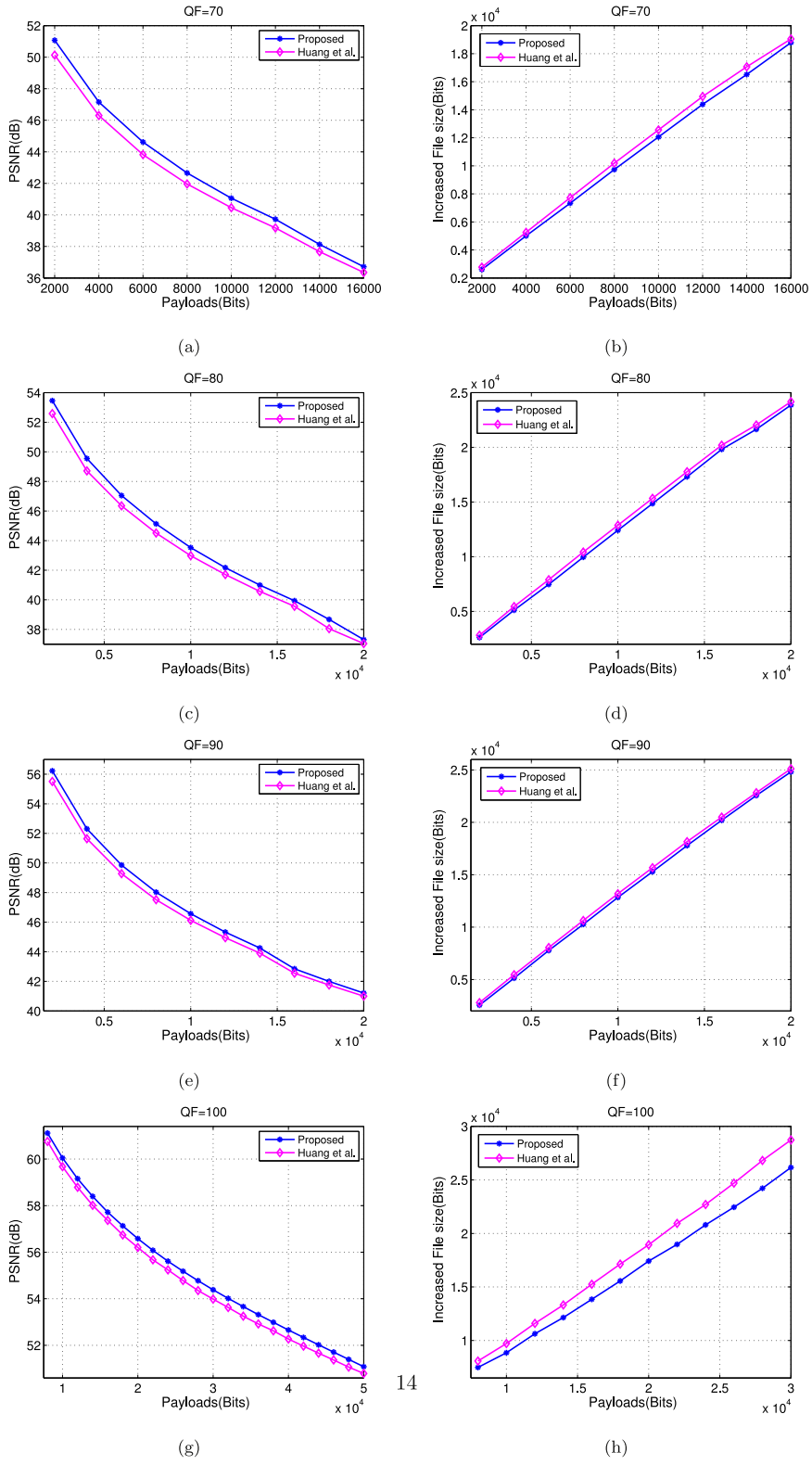


Fig. 5. Average PSNR values and increased file sizes under different embedding payloads.

in the lossless formats and be compressed into JPEG versions under different *QF*s, and then experiments are carried on these compressed JPEG images. As discussed in Section 3.1, we select host quantified DCT coefficients for embedding from K frequencies with the minimum unit distortions. A suitable K can well decrease the distortion of marked image for the proposed method. The experimental results with Lena as test image by setting different K values are given in Fig. 4, from which we see that the PSNR value of marked image reaches the peak when K is less than 63 and usually far less 63. That is to say frequency selection in quantification table is rather meaningful in improving the quality of marked image. After the optimal K frequencies are selected, we record them as one

part of payloads to be embedded, and then the receiver can relocate these selected K frequencies. To speed the processes of finding a suitable K, we can initialize $K = 30$ and then search the local optimum value toward two directions with the step 3.

We compare the proposed method with Wang et al.'s work [21] and Huang et al.'s work [22] on typical test images shown in Fig. 3 firstly. The comparative results are listed in Table 1. Note that, the data in bracket after Wang et al.'s work, Huang et al.'s work and the proposed method are PSNR and increased file size with the units dB and bits respectively. Although Wang et al.'s method can achieve high PSNR, their method will also result in serious bitstream expansion. What is more, Wang et al.'s method can not be applied to JPEG image with $QF = 100$ due to that the corresponding quantification table can not be modified. Huang et al.'s work [22] is the state-of-the-art RDH algorithm in JPEG image, and the proposed method outperforms it in both visual quality and file size of generated marked JPEG image.

The presented method can be seen as an improvement of Huang et al.'s method [22]. To verify the advantages of the presented method, we test it and Huang et al.'s method on database [23] with 96 images, and the average results generated by such two methods are figured in Fig. 5. Note that, we will discard some test images which can not provide enough capacities to be plotted entirely in Fig. 5. The proposed method can achieve the same embedding capacity as Huang et al.'s method due to the same HS-based embedding manner, thus the discarded test images will be the same in such two methods. From Fig. 5 we see that, comparing with Huang et al.'s method the improved average PSNR obtained by the proposed method is about 1 dB when QF are 70 and 80, and 0.5 dB when QF are 90 and 100. At the same time, the increased file sizes obtained by the proposed method are always smaller than those of Huang et al.'s method, and our advantages in bitstream expansion will be more obvious when $QF = 100$. The reasons of our advantages are mainly about two aspects: one is that frequencies are considered when selecting host quantified DCT coefficients for embedding, and another one is that an advanced block selection strategy is applied.

5. Conclusions

In this paper, we estimate unit distortion for each frequency of AC coefficient, and then utilize block selection strategy to modify blocks yielding less simulated distortions preferentially. The presented method outperforms Huang et al.'s work [22]. However, compared with Wang et al.'s method [21], although the bitstream expansion is reduced greatly by the proposed method, the visual distortion of marked image created by the proposed method is still poorer. In the future, we will continue improve the RDH scheme in JPEG image by modifying the quantification table as considered in Wang et al.'s method [21].

Acknowledgments

The authors thank the anonymous reviewers for their valuable comments. In order to facilitate the readers, we will post the Matlab implementation of this paper on our website: http://home.ustc.edu.cn/%7Ehoudd.

References

[1] A. Khan, A. Siddiqa, S. Munib, S.A. Malik, A recent survey of reversible watermarking techniques, Inf. Sci. 279 (2014) 251–272.
[2] J. Fridrich, M. Goljan, Lossless data embedding for all image formats, in: SPIE Proceedings of Photonics West, Electronic Imaging, Security and Watermarking of Multimedia Contents, volume 4675, 2002, pp. 572–583. San Jose
[3] J. Tian, Reversible data embedding using a difference expansion, IEEE Trans. Circuits Syst. Video Technol. 13 (8) (2003) 890–896.
[4] Z. Ni, Y. Shi, N. Ansari, S. Wei, Reversible data hiding, IEEE Trans. Circuits Syst. Video Technol. 16 (3) (2006) 354–362.
[5] P. Tsai, Y.C. Hu, H.L. Yeh, Reversible image hiding scheme using predictive coding and histogram shifting, Signal Process. 89 (2009) 1129–1143.
[6] V. Sachnev, H.J. Kim, J. Nam, S. Suresh, Y. Shi, Reversible watermarking algorithm using sorting and prediction, IEEE Trans. Circuits Syst. Video Technol. 19 (7) (2009) 989–999.
[7] X. Li, B. Yang, T. Zeng, Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection, IEEE Trans. Image Process. 20 (12) (2011) 3524–3533.
[8] B. Ou, X. Li, Y. Zhao, R. Ni, Y.Q. Shi, Pairwise prediction-error expansion for efficient reversible data hiding, IEEE Trans. Image Process. 22 (12) (2013) 5010–5021.
[9] X. Li, J. Li, B. Li, et al., High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion, Signal Process. 93 (2013) 198–205.
[10] X. Hu, W. Zhang, X. Li, N. Yu, Minimum rate prediction and optimized histograms modification for reversible data hiding, IEEE Trans. Inf. Forensics Secur. 10 (3) (2015) 653–664.
[11] B.G. Mobasseri, R.J. Berger, M.P. Marcinak, Y.J. NaikRaikar, Data embedding in JPEG bitstream by code mapping, IEEE Trans. Image Process. 19 (4) (2010) 958–966.
[12] Z. Qian, X. Zhang, Lossless data hiding in JPEG bitstream, J. Syst. Softw. 85 (2012) 309–313.
[13] Y. Hu, K. Wang, Z.M. Lu, An improved VLC-based lossless data hiding scheme for JPEG images, J. Syst. Softw. 86 (2013) 2166–2173.
[14] G. Xuan, Y.Q. Shi, Z. Ni, P. Chai, X. Cui, X. Tong, Reversible data hiding for JPEG images based on histogram pairs, in: Proc. Int. Conf. on Image Analysis and Recognition, Montreal, Canada, 2007, pp. 715–727.
[15] C.C. Chang, C.C. Lin, C.S. Tseng, W.L. Tai, Reversible hiding in DCT-based compressed images, Inf. Sci. 177 (13) (2007) 2768–2786.
[16] H. Sakai, M. Kuribayashi, M. Morii, Adaptive reversible data hiding for JPEG images, in: Proc. Int. Symp. on Inf. Theory and its Applications, Auckland, New Zealand, 2008, pp. 1–6.
[17] C.C. Lin, P.F. Shiu, DCT-based reversible data hiding scheme, in: Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication, ACM, pp. 327–335.
[18] Q. Li, Y. Wu, F. Bao, A reversible data hiding scheme for JPEG images, in: Proc. 11th Pacific Rim Conf. on Multimedia, in: Lecture Notes in Computer Science, 6297, 2010, pp. 653–664.
[19] X. Zhang, S. Wang, Z. Qian, et al., Reversible fragile watermarking for locating tampered blocks in JPEG images, Signal Process. 90 (2010) 3026–3036.
[20] T. Efimushkina, K. Egiazarian, M. Gabbouj, Rate-distortion based reversible watermarking for JPEG images with quality factors selection, in: Proc. 4th European Workshop on Vis. Inf. Process., Paris, France, 2013.
[21] K. Wang, Z.M. Lu, Y.J. Hu, A high capacity lossless data hiding scheme for JPEG images, J. Syst. Softw. 86 (2013) 1965–1975.
[22] F. Huang, X. Qu, H.J. Kim, J. Huang, Reversible data hiding in JPEG images, IEEE Trans. Circuits Syst. Video Technol. 26 (9) (2016) 1610–1621.
[23] Related images of the experiments, [Online]. Available: http://decsai.ugr.es/cvg/dbimagenes/g512.php.