

Reversible Data Hiding in Block Truncation Coding Compressed Images Using Quantization Level Swapping and Shifting

Wien Hong^{1,2*}, Shuozhen Zheng², Tung-Shou Chen³, Chien-Che Huang³

¹School of Computer and Software

Nanjing University of Information Science and Technology, China

²Department of Electronic Communication and Software Engineering

Nanfeng College of Sun Yat-Sen University, Guangzhou, China

³National Taichung University of Science and Technology

Department of Computer Science and Information Engineering, Taichung, Taiwan

*Corresponding author: wienhong@gmail.com

*Received January 2, 2016; revised March 24, 2016; accepted April 20, 2016;
published June 30, 2016*

Abstract

The existing reversible data hiding methods for block truncation coding (BTC) compressed images often utilize difference expansion or histogram shifting technique for data embedding. Although these methods effectively embed data into the compressed codes, the embedding operations may swap the numerical order of the higher and lower quantization levels. Since the numerical order of these two quantization levels can be exploited to carry additional data without destroying the quality of decoded image, the existing methods cannot take the advantages of this property to embed data more efficiently. In this paper, we embed data by shifting the higher and lower quantization levels in opposite direction. Because the embedment does not change numerical order of quantization levels, we exploit this property to carry additional data without further reducing the image quality. The proposed method performs no-distortion embedding if the payload is small, and performs reversible data embedding for large payload. The experimental results show that the proposed method offers better embedding performance over prior works in terms of payload and image quality.

Keywords: BTC, Difference expansion, Histogram shifting, Quantization levels.

1. Introduction

The advance of digital information has made our daily lives more convenient. However, it also has brought new challenges in security issues. Due to the rapid development of transmission techniques and the wide spread use of internet, the security issues, including network security [1]-[4] and multimedia security [5]-[6], have been increasingly receiving a great deal of attention from the research community. As a research area of multimedia security, data hiding methods have been extensively investigated for the purpose of authentication, identification, and privacy protection during the past decade. Data hiding is a technique for secret transmission by embedding messages into carriers. Many digital media can be used for carrying data and the digital images are perhaps the most commonly used carriers because they provide rich redundancies for data embedment. Moreover, digital images are often transmitted over internet and thus they are very suitable to serve as carriers. In general, a digital image used for embedding data is called a cover image and an image with data embedded is called a stego image. Since the data embedment is done by modifying the pixel values in the cover image, the distortions occur inevitably. To evade the detection by steganalyzers [7]-[8], either visually or statistically, the distortion should be as small as possible. As a result, many data hiding methods adopt machine learning [9]-[10] or image processing [11]-[12] techniques to reduce the distortion further, or to keep the distortion imperceptible by considering the human visual system. For a data hiding method, a large payload is usually desirable. However, a large payload often causes larger distortion. As a result, to keep a balance between the payload and image quality is a challenging task for a data hiding method.

According to reversibility of the cover image, the data hiding methods can be classified into two categories: non-reversible [13]-[14] and reversible [15]-[18]. Non-reversible data hiding method distorts the cover images permanently and therefore, the produced stego image cannot be restored to its original state. The well-known LSB replacement, LSB matching [13] and adaptive pixel pair matching (APPM) [14] are of this type. On the other hand, reversible data hiding (RDH) method recovers the original image from the distorted, stego image after extracting the embedded data. Therefore, it is suitable for applications where no distortion is allowed such as medical or military images.

Ni et al. [15] proposed a RDH method using histogram shifting. This method vacates a histogram bin that is next to the peak bin, and data bits are embedded by shifting the pixels in the peak bin. Many RDH methods are developed based on Ni et al.'s method. For example, Li et al. [16] proposed a RDH method using pixel value ordering to efficiently embed data by shifting the smallest and largest pixels of a sorted block. Wang et al. [17] extended Li et al.'s work by using a dynamic pixel block partition scheme and provided a better performance than [16]. Hong et al. [18] proposed a RDH method using Delaunay triangulation and selective embedment. In their method, a selection mechanism is used to control the embedding regions in the cover image to evade the detection of steganalyzers.

The aforementioned methods embed data in the spatial domain of cover image. However, due to the limited transmission bandwidth and storage capacity, many digital images are transmitted or stored in a compressed format. As a result, the traditional RDH methods cannot directly be applied to images in compressed format. Some RDH methods have been proposed so far for various compressed domain, such as vector quantization (VQ) [19], JPEG [20], and block truncation coding (BTC) compressed images [21]-[25], to accommodate different requirements for various applications. Since BTC [26] simply represents an image block by

two quantization levels (QLs) and a bitmap, it requires insignificant computational cost but also offers acceptable image quality and compression rate. As a result, it is very suitable for applications with limited computational, energy, and storage resources, such as mobile or remote censoring devices.

Till now, only a few RDH methods for BTC compressed images (or BTC codes) are proposed. Hong et al. [21] proposed a RDH in BTC codes based on prediction and histogram shifting. This method partitioned higher and lower QLs into blocks, and then a reversible method is applied to these blocks for data embedment. Chen et al. [22] proposed an interesting no-distortion embedding method for BTC codes in that the decoded stego image is identical to the original BTC compressed image. Normally, the higher QLs should be larger than or equal to the corresponding lower QLs. However, decoding the swapped QLs with a flipped bitmap also obtains an identical image block. Chen et al. exploited this interesting property to embed data by swapping QLs and flipping bitmap to produce a no-distortion stego image. Chang et al. [23] also proposed a reversible method for BTC codes using side match. Although this method achieves higher payload than that of [21] and [22], it cannot be decoded by a standard BTC decoder and thus the application is limited. Lo et al. [24] made use of histogram shifting technique to embed secret data into the QLs and the embedded codes still follows the standard BTC format. However, the payload of this method is limited. Tsai et al. [25] proposed an efficient reversible data hiding algorithm for BTC compressed images based on difference expansion and median edge detector. The QLs are predicted, and data are embedded by expanding the prediction errors. If the numerical order of the QLs is swapped when trying to embed data, this method changes the recording positions of these two QLs and flips the corresponding bitmap to signal the decoder for a correct reconstruction. Although Tsai et al.'s method achieves lower bit rate and higher payload than those of [22] and [24], the swapping of QLs can no longer be used for carrying additional data bits using the method proposed in [22] because it has been used for indicating embeddable and un-embedding blocks.

In this paper, we proposed a RDH method by shifting the QLs in opposite direction. The QLs with the most frequently occurring difference are shifted and data bits are embedded into the shifted histogram. Because the shifting operation preserves the numerical order of QLs, the swapping of QLs can still be employed to carry an additional data bit with no-distortion. Besides, a threshold is used to prevent those likely un-embeddable differences from being shifted. Therefore, the proposed method achieves significantly better embedding performance than previous works, especially at low embedding rate.

The rest of this paper is organized as follows. Section 2 briefly describes BTC and Tsai et al.'s methods. Section 3 gives the algorithm of the proposed work. Section 4 shows the experimental results and concluding remarks are given in Section 5.

2. Related Work

In this section, we briefly introduce the BTC compression algorithm and Tsai et al.'s RDH method for BTC compressed codes.

2.1 BTC compression technique

The BTC algorithm [26] is made use of averaged pixel values of blocks as the QLs and use a bitmap to record which QLs should be used at the corresponding position. The detailed compressed algorithm is shown as follows. Firstly, BTC partitions the cover image I into N

blocks $\{Bk_i\}_{i=1}^N$ of size $u \times u$. Let $\{p_{t,i}\}_{t=1}^{u \times u}$ be the pixels in block Bk_i and $B_i = \{b_{t,i}\}_{t=1}^{u \times u}$ be a binary bitmap. To encode Bk_i , the mean value m_i of Bk_i is calculated. For those pixels in Bk_i with value smaller than or equal to m_i are averaged and rounded to obtain a lower QL $q_{L,i}$. Similarly, pixels with value larger than m_i are averaged and rounded to obtain a higher QL $q_{H,i}$. Once the QLs are obtained, BTC made use of bitmap $B_i = \{b_{t,i}\}_{t=1}^{u \times u}$ to record the QLs for pixels $\{p_{t,i}\}_{t=1}^{u \times u}$. If $p_{t,i} \leq m_i$, $b_{t,i}$ is set to 0; otherwise, $b_{t,i}$ is set to 1. As a result, the BTC codes for i -th block is a trio $(q_{L,i}, q_{H,i}, B_i)$ and $\{q_{L,i}, q_{H,i}, B_i\}_{i=1}^N$ is the BTC codes for image I . Notice that $q_{L,i} \leq q_{H,i}$ always holds for each block. To decode a BTC compressed code $(q_{L,i}, q_{H,i}, B_i)$, we simply replace bits '0' in B_i with $q_{L,i}$ and replace bit '1' in B_i with $q_{H,i}$. Each block is performed in the same manner and a BTC compressed image can be reconstructed.

2.2 Tsai et al.'s method

In 2013, Tsai et al. [25] proposed a RDH method for BTC compressed images based on difference expansion and median edge detector (MED). In their method, QLs are predicted, and k -bit data are embedded into the expanded prediction errors. After embedment, the order of QLs of some blocks is possibly changed and this will lead to incorrect data extraction and block reconstruction. Therefore, this method classifies blocks into groups of embeddable and un-embeddable. For those un-embeddable blocks, the lower and higher QLs are inversely recorded, and thus the location map is not required.

Here is a brief description about Tsai et al.'s method. Let the BTC compressed codes be $\{q_{L,i}, q_{H,i}, B_i\}_{i=1}^N$. To embed data into block Bk_i , $q_{L,i}$ is firstly predicted by MED predictor to obtain the predicted value $\hat{q}_{L,i}$:

$$\hat{q}_{L,i} = \begin{cases} \min(q_{L,i}^w, q_{L,i}^n) & q_{L,i}^{nw} \geq \max(q_{L,i}^w, q_{L,i}^n), \\ \max(q_{L,i}^w, q_{L,i}^n) & q_{L,i}^{nw} \leq \min(q_{L,i}^w, q_{L,i}^n), \\ q_{L,i}^w + q_{L,i}^n - q_{L,i}^{nw} & \text{otherwise,} \end{cases} \quad (1)$$

where $q_{L,i}^w, q_{L,i}^n$ and $q_{L,i}^{nw}$ are the west, north, and north-west lower QLs of the current lower QL $q_{L,i}$, respectively. The prediction value of $q_{H,i}$, denoted by $\hat{q}_{H,i}$, can be obtained in the similar way. Let $d_{L,i} = |\hat{q}_{L,i} - q_{L,i}|$ and $d_{H,i} = |\hat{q}_{H,i} - q_{H,i}|$ be the prediction errors. If one of or both $d_{L,i}$ and $d_{H,i}$ are smaller than or equal to a predefined embedding parameter EP , block Bk_i is classified as embeddable and otherwise, un-embeddable. To embed k -bit data s_L^k into the lower QL of an embeddable block Bk_i , s_L^k is transformed into its decimal representation $(s_L^k)_{10}$, and $(s_L^k)_{10}$ is then embedded by expand the prediction error:

$$q'_{L,i} = \begin{cases} \hat{q}_{L,i} + 2^k \times d_{L,i} + (s_L^k)_{10} & \hat{q}_{L,i} \leq q_{L,i}, \\ \hat{q}_{L,i} - 2^k \times d_{L,i} - (s_L^k)_{10} & \text{otherwise.} \end{cases} \quad (2)$$

Similarly, another k -bit s_H^k can be embedded into the higher QL $q_{H,i}$ and obtain $q'_{H,i}$. After embedment, the modified compressed code $(q'_{L,i}, q'_{H,i}, B_i)$ of block Bk_i is $(q'_{L,i}, q'_{H,i}, B_i)$. However, the QLs $q'_{L,i}$ and $q'_{H,i}$ are possibly overflow or underflow, or the order of them is swapped, i.e., $q'_{L,i} > q'_{H,i}$. In these cases, blocks Bk_i become un-embeddable. If this happened, the encoder output the codes $(q'_{L,i}, q'_{H,i}, B_i) = (q_{H,i}, q_{L,i}, \bar{B}_i)$ to signal the decoder it is an

un-embeddable block. All the blocks are encoded using the same manner to obtain the stego BTC codes $\{q'_{1,i}, q'_{2,i}, B'_i\}_{i=1}^N$.

To extract the embedded data, the embeddable and un-embeddable blocks are classified by inspecting $q'_{1,i}$ and $q'_{2,i}$. If $q'_{1,i} \leq q'_{2,i}$, k bits are embedded in $q'_{1,i}$ and $q'_{2,i}$, respectively. In this case, the predicted value $\hat{q}'_{1,i}$ and $\hat{q}'_{2,i}$ are obtained first, as in the embedding procedures. The embedded $(s_L^k)_{10}$ can be extracted by

$$(s_L^k)_{10} = |\hat{q}'_{1,i} - q'_{1,i}| \bmod 2^k. \quad (3)$$

The original quantization level $q_{L,i}$ can be recovered by using the following equation:

$$q_{L,i} = \begin{cases} \hat{q}'_{1,i} + \lfloor (\hat{q}'_{1,i} - q'_{1,i}) / 2^k \rfloor & \text{if } \hat{q}'_{1,i} \leq q'_{1,i}, \\ \hat{q}'_{1,i} - \lfloor (\hat{q}'_{1,i} - q'_{1,i}) / 2^k \rfloor & \text{otherwise.} \end{cases} \quad (4)$$

The embedded data $(s_H^k)_{10}$ can be extracted and $q_{H,i}$ can be recovered in the similar way. If $q'_{1,i} > q'_{2,i}$, no data are embedded and the original trio can be recovered by $q_{L,i} = q'_{2,i}$, $q_{H,i} = q'_{1,i}$ and the original bitmap is \bar{B}'_i .

3. Proposed method

As mentioned in the introduction, decoding a BTC trio $(q_{L,i}, q_{H,i}, B_i)$ or $(q_{H,i}, q_{L,i}, \bar{B}_i)$ gives an identical image block. As a result, utilizing the numerical order of $q_{L,i}$ and $q_{H,i}$ not only embeds one additional bit information but also produces a no-distortion stego block. In Tsai et al.'s method [25], the numerical order of QLs is used alternatively to classify blocks into embeddable and un-embeddable. That is, if the block is un-embeddable, the method swaps the values of higher and lower QLs. Although this operation successfully signal the decoder for a correct data extraction, the numerical order can no longer be used to hold one secret bit in a no-distortion way. To see how the swapped QLs are distributed, we use Fig. 1 to show a map for the Lena image after applying Tsai et al.'s method at $EP=8$ with $k=1$ and $k=2$, respectively. In this figure, gray dots represent k -bit data are successfully embedded (i.e., embeddable). White dots and black dots represent un-embeddable blocks, where white dots are those blocks with prediction errors larger than EP and black dots are those QLs with numerical order are changed after data embedment (i.e., $q'_{L,i} > q'_{H,i}$).

Tsai et al.'s method use $k=2$ to acquire more embedding capacity. However, as shown in Figs. 1(a) and (b), setting $k=2$ also induces more black dots and thus the payload is penalized because more blocks become un-embeddable.

In this section, we proposed a RDH method for BTC compressed images by iteratively shifting the higher and lower QLs with the most frequently occurring difference between them. Since the higher QLs are shifted toward 255 and lower QLs are shifted toward 0, the numerical order of QLs is preserved. Therefore, the order can be employed to carry an additional data bit with no-distortion. A threshold mechanism is also adopted to exclude those likely un-embeddable QLs from being shifted. The detailed embedding techniques of the proposed work are given in the following sub-section.

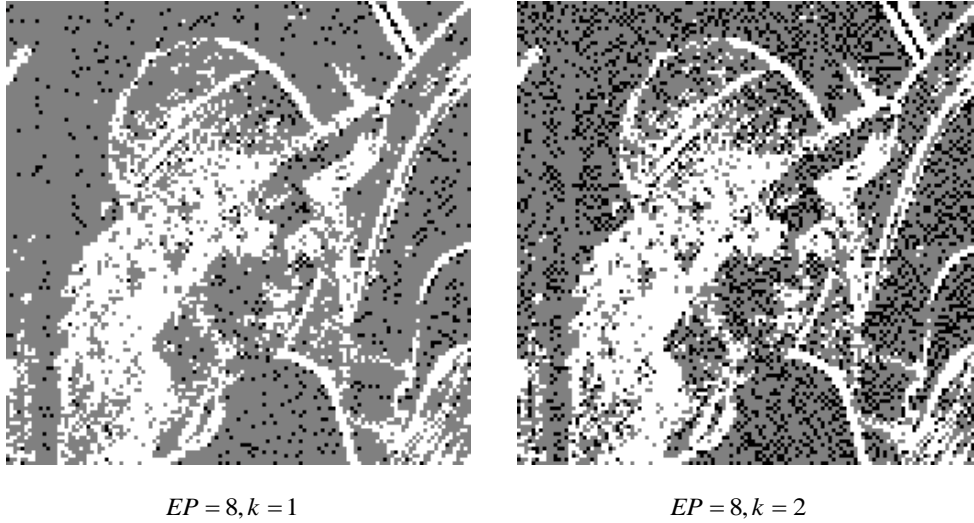


Fig. 1. Distribution of swapped QLs for the Lena image after applying Tsai et. al.'s method.

3.1 Preprocessing of the quantization levels

Since the shifting of QLs may result in overflow and underflow, we need to preprocess them to ensure no QL after embedding is out of the range $[0,255]$. Since the proposed method can perform multi-level embedding, we denote ℓ the desirable embedding levels. Let trios $\{q_{L,i}^o, q_{H,i}^o, B_i\}_{i=1}^N$ be the original BTC codes. $q_{L,i}^o$ and $q_{H,i}^o$ are then preprocessed using the following manner: If $q_{H,i}^o > 255 - \ell$, set $q_{H,i} = q_{H,i}^o - \ell$ and $q_{L,i} = q_{L,i}^o - \ell$. If $q_{L,i}^o < \ell$, set $q_{L,i} = q_{L,i}^o + \ell$ and $q_{H,i} = q_{H,i}^o + \ell$. Each block is processed in the same manner and the preprocessed BTC codes $\{q_{L,i}, q_{H,i}, B_i\}_{i=1}^N$ are obtained. Since the decoder requires the location of the pre-processing QLs, a binary location map LM is used to record the pre-shifting locations. LM is then compressed using JBIG 2 codec to obtain a compressed location map cLM .

3.2 QL-shifting embedding

QL-shifting embeds data by shifting the QLs with difference equal to the most frequently occurring differences between the higher and lower QLs. It is known that most natural images possess a property that the smooth regions are more than complex regions. As a result, the QL difference histogram of a BTC compressed natural image tends to be densely distributed at small difference and sparsely distributed at larger difference. Let $H_\alpha(\{q_{H,i} - q_{L,i}\}_{i=1}^N)$ be the difference histogram representing the number of elements valued α in the difference set $\{q_{H,i} - q_{L,i}\}_{i=1}^N$. **Figs. 2(a)** and **(b)** show the difference histogram of the Lena and Baboon images, respectively. Both figures reveal that the histogram peak ρ , defined by

$$\rho = \operatorname{argmax}_{0 \leq \alpha \leq 255} H_\alpha(\{q_{H,i} - q_{L,i}\}_{i=1}^N), \quad (5)$$

tends to be located at small difference.

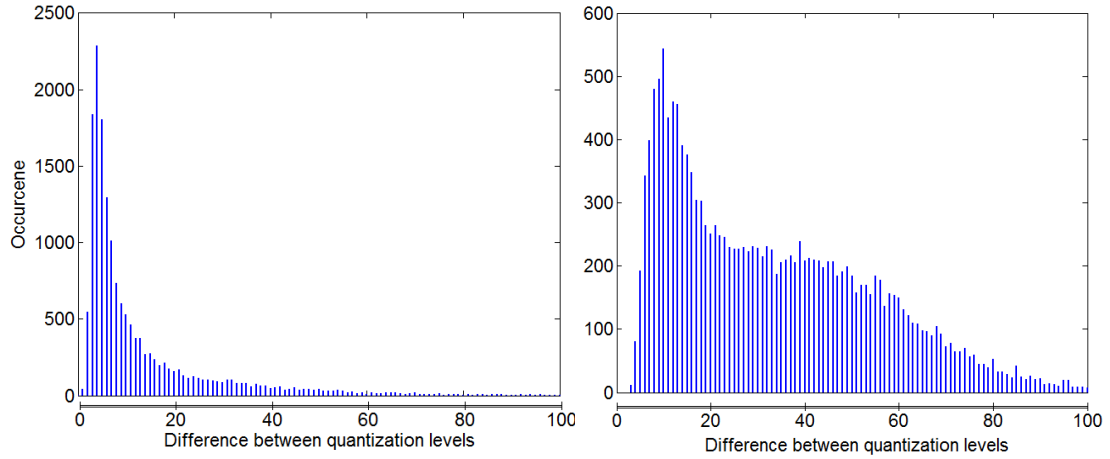


Fig. 2. Difference histogram of Lena and Baboon image

Our method is to choose the most frequently occurring difference ρ as the referential difference for data embedding. Let ρ_L be the referential difference ρ_L obtained from **Eq. (5)**. The lower QL $q_{L,i}$ is then modified for carrying a message bit s , or shifted by one unit using the following equation

$$q'_{L,i} = \begin{cases} q_{L,i} - s & q_{H,i} - q_{L,i} = \rho_L, \\ q_{L,i} - 1 & q_{H,i} - q_{L,i} > \rho_L, \\ q_{L,i} & \text{otherwise.} \end{cases} \quad (6)$$

All the lower QLs $\{q_{L,i}\}_{i=1}^N$ are processed in the same way and $H_{\rho_L}(\{q_{H,i} - q_{L,i}\}_{i=1}^N)$ message bits can be embedded. Let $\{q'_{L,i}, q_{H,i}, B_i\}_{i=1}^N$ be the processed codes. Once the lower QLs are processed, the higher QLs are processed to embed data in a similar way. That is, the referential difference ρ_H is obtained from the difference set $\{q_{H,i} - q'_{L,i}\}_{i=1}^N$, and the higher QLs are then visited and use the equation

$$q'_{H,i} = \begin{cases} q_{H,i} + s & q_{H,i} - q'_{L,i} = \rho_H, \\ q_{H,i} + 1 & q_{H,i} - q'_{L,i} > \rho_H, \\ q_{H,i} & \text{otherwise,} \end{cases} \quad (7)$$

to embed a message bit s or to shift the higher QLs by one unit. After embedding, the stego BTC codes $\{q'_{L,i}, q'_{H,i}, B'_i\}_{i=1}^N$ of blocks $\{Bk_i\}_{i=1}^N$ can be obtained, where $\{B'_i\}_{i=1}^N = \{B_i\}_{i=1}^N$. Note that $q'_{L,i} \leq q_{L,i}$ and $q'_{H,i} \geq q_{H,i}$ always hold. As a result, the numerical order of the QLs is preserved. Besides, the embedment using **Eqs. (6)** and **(7)** will not cause any overflow or underflow in that the modification is performed on the pre-processed image. As a result, the embedment of QLs does not require the information of the location map.

3.3 Threshold mechanism

Since the value of reference difference ρ_L (or ρ_H) for most images tends to be small, which imply that smaller $d_i = q_{H,i} - q_{L,i}$ has better chance to equal reference difference for carrying a data bit when QL-shifting embedding is applied. Therefore, the proposed method uses a threshold to select those likely embeddable QLs for data embedment, and those likely un-embeddable QLs are skipped from the QL-shifting operation to prevent degrading the stego image quality. For a BTC compressed image, neighboring differences d_i tend to have similar values. We thus use the visited differences to estimate whether the current difference d_i is possible for carrying a data bit or not. Let $\hat{\sigma}_i = \sigma(d_i^w, d_i^n, d_i^{nw})$ be the standard deviation of neighboring differences, where $\sigma(\cdot)$ is the standard deviation operator and d_i^w, d_i^n, d_i^{nw} are the differences to the west, north and north-west of the current difference d_i , respectively. If $\hat{\sigma}_i \leq T$, where T is a pre-defined threshold, d_i has better chance to equal ρ_L (or ρ_H), and thus we can use the QL-shifting embedding technique to perform data embedment, as described in the previous section. On the other hand, if $\hat{\sigma}_i > T$, the corresponding difference d_i is likely unequal to ρ_L (or ρ_H). As a result, we just skip the current visited QL $q_{L,i}$ (or $q_{H,i}$) and do not perform QL-shifting operation.

3.4 QL-swapping embedding

Since the QL-shifting embedding preserves the numerical order of QLs, the swapping of QLs can be employed to carry additional data bit in a no-distortion manner. We term this technique QL-swapping. QL-swapping can be applied before or after QL-shifting, but applied after QL-shifting will be easier for implementation. Let $\{q'_{L,i}, q'_{H,i}, B'_i\}_{i=1}^N$ be the stego BTC codes after applying QL-shifting embedding. To embed a message bit s using QL-swapping, if $s = 0_2$, block Bk_i is coded by $(q'_{L,i}, q'_{H,i}, B'_i)$. On the other hand, if $s = 1_2$, block Bk_i is coded by $(q'_{H,i}, q'_{L,i}, \bar{B}'_i)$. We use the trios $\{q''_{L,i}, q''_{H,i}, B''_i\}_{i=1}^N$ to represent the coded results. Because QL-swapping utilizes the numerical order of $q'_{L,i}$ and $q'_{H,i}$ to embed one bit, $q'_{L,i}$ and $q'_{H,i}$ must have distinct values. As a result, if there are N_{eq} blocks with identical QLs (i.e., $q'_{L,i} = q'_{H,i}$), the proposed method just skip these blocks and does not perform data embedment. Fortunately, a block with identical QLs occurs rarely for most natural images. Therefore, the skipping operation affects the payload insignificantly. Note that QL-swapping operation not only embeds $N - N_{eq}$ bits but also provides a BTC decoded image with no-distortion.

3.5 Embedding and extraction procedures

In this section, we describe the embedding, extraction and image recovery procedures in detail. Let $\{q^o_{L,i}, q^o_{H,i}, B_i\}_{i=1}^N$ be the BTC compressed codes of a cover image I , and S be the secret data. Given a threshold T and an embedding level ℓ , secret data S can be embedded using the following procedures. Firstly, the QLs $q^o_{L,i}$ and $q^o_{H,i}$ are pre-processed to obtain a compressed location map cLM , as described in Section 3.1. The pre-processed code is denoted by $\{q_{L,i}, q_{H,i}, B_i\}_{i=1}^N$. Concatenate cLM and S to form a message bit-stream M . Once the QLs are

preprocessed, we firstly embed the first $|M| - (N - N_{eq})$ bits by applying the QL-Shifting technique iteratively, starting from embedding level one to embedding level ℓ under the guidance of threshold T , as described in Sections 3.2 and 3.3.

For j -th embedding level, the referential differences $\rho_L^{(j)}$ and $\rho_H^{(j)}$ are calculated first, and the stego codes $\{q_{L,i}^{(j)}, q_{H,i}^{(j)}, B_i^{(j)}\}_{i=1}^N$ can be obtained after the message bits are embedded. The referential differences $\rho_L^{(j)}$ and $\rho_H^{(j)}$ are then converted to binary representations, and are prepended to the un-embedded messages bits. The prepended message bits are then embedded into stego codes $\{q_{L,i}^{(j)}, q_{H,i}^{(j)}, B_i^{(j)}\}_{i=1}^N$ in the next embedding level to obtain a new stego codes $\{q_{L,i}^{(j+1)}, q_{H,i}^{(j+1)}, B_i^{(j+1)}\}_{i=1}^N$. These processes are repeated until all $|M| - (N - N_{eq})$ bits are embedded, and the final stego code is $\{q_{L,i}^{(\ell)}, q_{H,i}^{(\ell)}, B_i^{(\ell)}\}_{i=1}^N$, where ℓ is the last embedding level. The remaining $N - N_{eq}$ bits in M are then embedded using the QL-swapping technique, as described in section 3.4. The final stego codes $\{q_{L,i}^{(\ell)}, q_{H,i}^{(\ell)}, B_i^{(\ell)}\}_{i=1}^N$ can then be obtained. The embedding procedures of each step are depicted in Fig. 3.

Note that it is also possible to embed data into BTC codes by firstly applying the QL-swapping technique instead of QL-shifting. However, embedment performed in this way requires additional computation cost because the original QLs should be recovered in advanced so that QL-shifting can be correctly performed. As a result, the proposed method embeds data using the QL-shifting before QL-swapping.

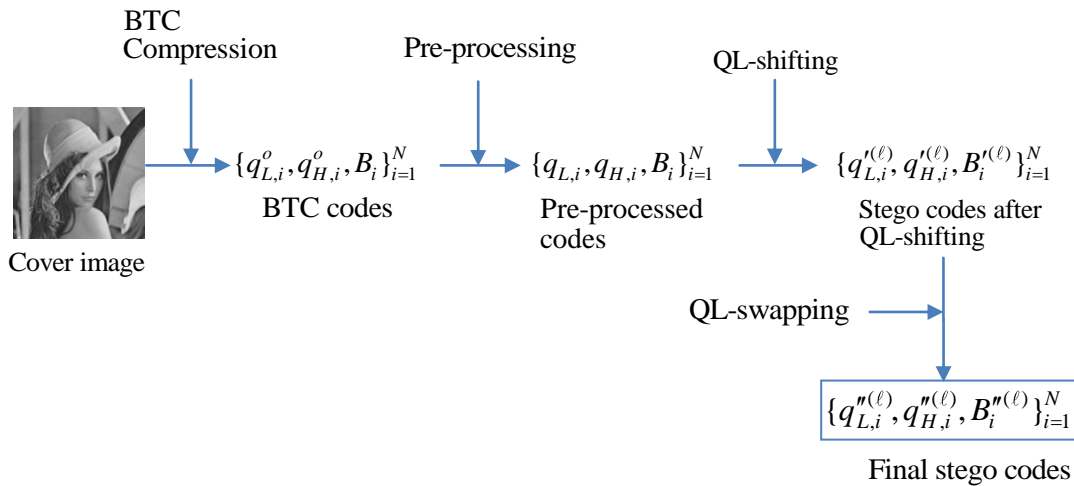


Fig. 3. Embedding procedures of the proposed method.

The referential differences $\rho_L^{(\ell)}$ and $\rho_H^{(\ell)}$ of the last embedding level are outputted as one of the keys for data extraction and image recovery. The keys, including referential differences $\rho_L^{(\ell)}, \rho_H^{(\ell)}$, threshold T , the size of the compressed location map $|cLM|$, and the size of secret data $|S|$ are transmitted to the receiver side via a secret channel. For a cover image of size 512×512 with block size 4×4 , we use 8 bits each to record $\rho_L^{(\ell)}, \rho_H^{(\ell)}$ and T , and use 16 bits each to record $|cLM|$ and $|S|$. Therefore, the key size of the proposed method is $8 \times 3 + 16 \times 2 = 56$ bits. It is interesting to note that if the payload is smaller than $N - N_{eq}$, only

QL-swapping embedding is needed to be applied. In this case, the decoded BTC image is identical to the original BTC image.

The original BTC codes can be recovered and embedded secret data can be successfully extracted if the receiver has the stego codes $\{q_{L,i}^{n(\ell)}, q_{H,i}^{n(\ell)}, B_i^{n(\ell)}\}_{i=1}^N$ and the keys $\rho_L^{(\ell)}, \rho_H^{(\ell)}, T, |cLM|$ and $|S|$. We firstly extract $N - N_{eq}$ bits by decoding $\{q_{L,i}^{n(\ell)}, q_{H,i}^{n(\ell)}, B_i^{n(\ell)}\}_{i=1}^N$ using QL-swapping operation, and the stego codes $\{q_{L,i}^{r(\ell)}, q_{H,i}^{r(\ell)}, B_i^{r(\ell)}\}_{i=1}^{N \times N}$ are obtained. We then use the keys $\rho_L^{(\ell)}, \rho_H^{(\ell)}$, and T to extract the messages embedded in the last embedding level ℓ that was embedded by using QL-shifting and to recover the stego codes $\{q_{L,i}^{r(\ell-1)}, q_{H,i}^{r(\ell-1)}, B_i^{r(\ell-1)}\}_{i=1}^N$. The referential differences $\rho_L^{(\ell-1)}$ and $\rho_H^{(\ell-1)}$ can be obtained from the extracted messages, and the messages embedded in level $\ell - 1$ can be extracted. Repeat this process ℓ times and the stego codes $\{q_{L,i}, q_{H,i}, B_i\}_{i=1}^N$ can be recovered. Concatenate the extracted message, the compressed location map cLM and the embedded messages S can be obtained with the aid of $|cLM|$ and $|S|$. Decompress cLM to obtain the location map LM , the pre-shifted quantization levels can be restored and thus, the original BTC codes $\{q_{L,i}^o, q_{H,i}^o, B_i\}_{i=1}^N$ are obtained.

3.6 A simple example

In this section, we use a simple example to illustrate the proposed method. For simplicity, the pre-processing procedures and the embedding of location map is not considered. Besides, this example only illustrates a single layer embedding and the threshold mechanism is not used. In this example, we use a subscript to refer to an element's position in row-major order. For example, the position of an element located at r -th row and c -th column in a $R \times C$ matrix is indexed by $R \times (r - 1) + c$.

To begin with, let $\{q_{L,i}, q_{H,i}, B_i\}_{i=1}^9$ be the pre-processed BTC codes and $S = \{s_m\}_{m=1}^{12} = 110100010101_2$ be the secret data. $\{q_{L,i}\}_{i=1}^9$ and $\{q_{H,i}\}_{i=1}^9$ are shown in **Figs. 4 (a)** and **(b)**, respectively. Since $N_{eq} = 0$, $S = M$ and $|S| = 12$, we embed the first 3 bits by QL-shifting, and embed the last 9 bits using QL-swapping. To embeds $\{s_m\}_{m=1}^3 = 110_2$ using QL shifting, the differences $\{d_i\}_{i=1}^9 = \{q_{H,i} - q_{L,i}\}_{i=1}^9 = \{1, 3, 4, 3, 2, 5, 8, 6, 7\}$ are calculated. The referential difference $\rho_L = 3$ is then obtained and two bits can be embedded in $\{q_{L,i}\}_{i=1}^9$ since there are two elements in $\{d_i\}_{i=1}^9$ valued 3. The first difference is $q_{H,1} - q_{L,1} = 1 < \rho_L$, according to **Eq. (6)**, we simply set $q'_{L,1} = q_{L,1} = 13$. The second difference is $q_{H,2} - q_{L,2} = 12 - 9 = 3 = \rho_L$, therefore, a bit can be embedded. Extract $s_1 = 1_2$ from S , and set $q'_{L,2} = q_{L,2} - s_1 = 9 - 1 = 8$. Similarly, since the third difference $q_{H,3} - q_{L,3} = 9 - 5 = 4 > \rho_L$, we set $q'_{L,3} = q_{L,3} - 1 = 5 - 1 = 4$.

The fourth difference is $q_{H,4} - q_{L,4} = 15 - 12 = \rho_L$, therefore, we extract $s_2 = 1_2$ from S and set $q'_{L,4} = q_{L,4} - s_2 = 12 - 1 = 11$. The remaining lower QLs $\{q_{L,i}\}_{i=5}^9$ can be processed in the similar way and thus $\{q'_{L,i}\}_{i=1}^9$ can be obtained (see **Fig. 4(c)**). To embed the last bit $s_3 = 0_2$ using QL-shifting, we calculate $\{d_i\}_{i=1}^9 = \{q_{H,i} - q'_{L,i}\}_{i=1}^9 = \{1, 4, 5, 4, 2, 6, 9, 7, 8\}$ and thus obtain $\rho_H = 4$. Since the first difference is 1, we set $q'_{H,1} = q_{H,1} = 14$. The second difference is 4, we thus embed $s_3 = 0_2$ by setting $q'_{H,2} = q_{H,2} + s_3 = 12$. Now all $\{s_i\}_{i=1}^3$ are embedded and thus the modification of the remaining QLs is not required. Therefore, we set

$\{q'_{H,i}\}_{i=3}^9 = \{q_{H,i}\}_{i=3}^9$. The resultant QLs $\{q'_{H,i}\}_{i=1}^9$ are shown in **Fig. 4(d)**. Therefore, the stego BTC codes after QL-shifting is $\{q'_{L,i}, q'_{H,i}, B'_i\}_{i=1}^9$, where $B'_i = B_i$, for $1 \leq i \leq 9$.

After processed QL-shifting embedding, we perform the QL-swapping embedding for the last 9 bits. Because $s_4 = 1_2$, we swap the values of $q'_{L,1}$ and $q'_{H,1}$, and flip the bitmap B'_1 . Therefore, $q''_{L,1} = q'_{H,1} = 14$, $q''_{H,1} = q'_{L,1} = 13$, and $B''_1 = \overline{B'_1}$. Similarly, because $s_5 = 0_2$, we set $q''_{L,2} = q'_{L,2} = 8$, $q''_{H,2} = q'_{H,2} = 12$, and $B''_2 = B'_2$. The other 7 bits $\{s_m\}_{m=6}^{12}$ are embedded in the similar way. The final stego codes $\{q''_{L,i}, q''_{H,i}, B''_i\}_{i=1}^9$ are therefore obtained. We show $\{q''_{L,i}\}_{i=1}^9$ and $\{q''_{H,i}\}_{i=1}^9$ in **Figs. 4 (e) and (f)**, respectively, and the swapped QLs are marked by gray.

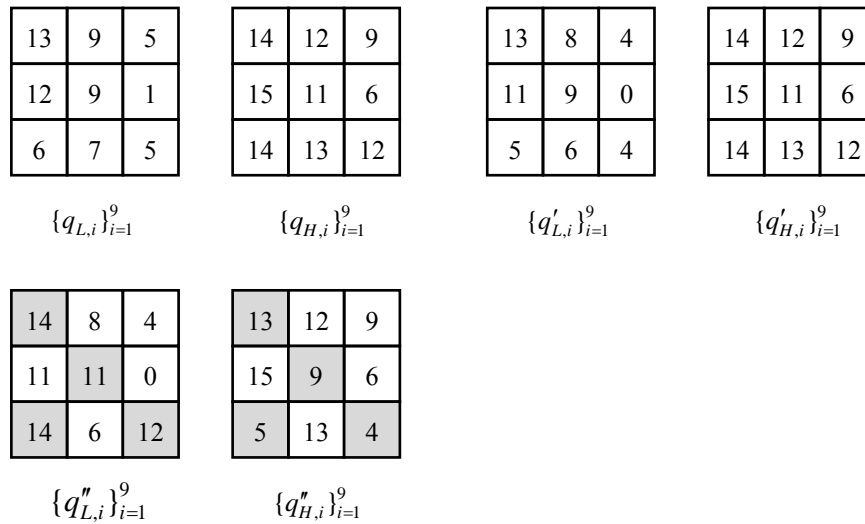


Fig. 4. Simple example of the proposed method.

To extract the embedded bits and recover the BTC codes, we firstly extract bits $\{s_m\}_{m=4}^{12}$ embedded by using QL-swapping, and recover the codes $\{q'_{L,i}, q'_{H,i}, B'_i\}_{i=1}^9$ from $\{q''_{L,i}, q''_{H,i}, B''_i\}_{i=1}^9$. The bits $s_3 = 0_2$ can be extracted from $\{q'_{H,i}\}_{i=1}^9$ and $\{q_{H,i}\}_{i=1}^9$ can be recovered with the aid of $\rho_H = 4$. Similarly, $\{s_m\}_{m=1}^2 = 11_2$ can be extracted from $\{q'_{L,i}\}_{i=1}^9$ and $\{q_{L,i}\}_{i=1}^9$ can be recovered with the aid of $\rho_L = 3$. The original BTC codes $\{q_{L,i}, q_{H,i}, B_i\}_{i=1}^9$ can finally be recovered.

4. Experimental Results

In this section, we performed several experiments to demonstrate the applicability of the proposed method and compare the performance with other methods. Eight 512×512 grayscale images shown in **Fig. 5**, including Lena, Jet, Peppers, Boat, Tiffany, Tank, Truck, and Baboon were used as the test images. These images can be obtained from USC-SIPI image database [27]. Secret data were generated by using a pseudo random number generator (PRNG). We use the peak signal-to-noise ratio (PSNR) to measure the stego image quality, which is defined by

$$PSNR = 10 \log_{10} \frac{255^2}{\frac{1}{N \times u^2} \sum_{i=1}^{N \times u^2} (I_i - I'_i)^2}, \quad (8)$$

where I_i and I'_i are the images decoded from the original BTC codes and the stego BTC codes, respectively. In general, a higher PSNR indicates that two images are more similar. In all the experiments, a block size of 4×4 is used.

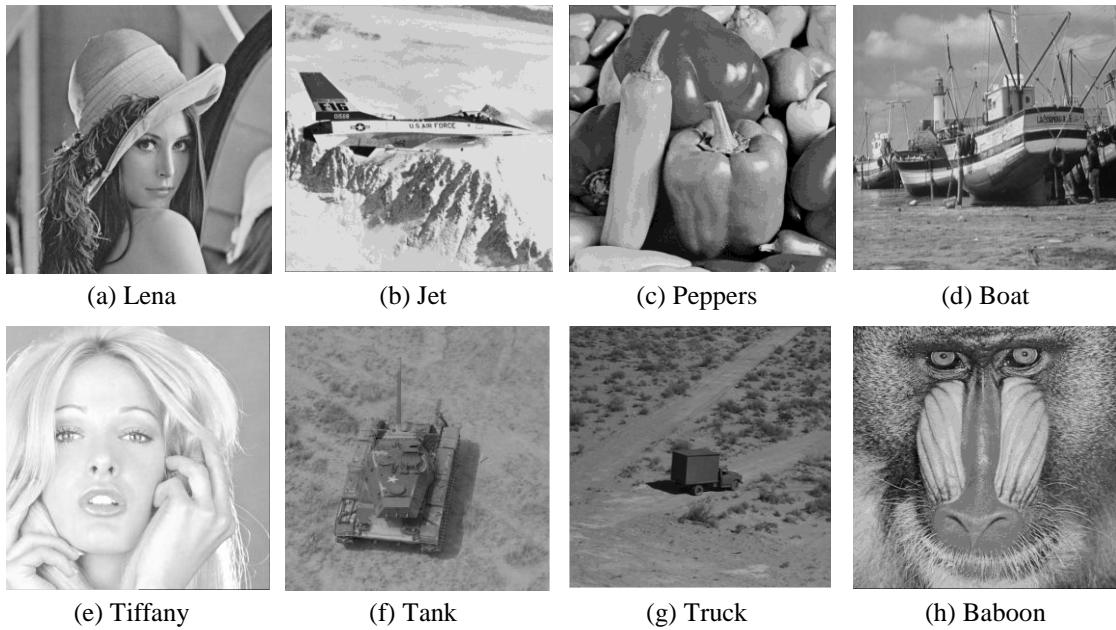


Fig. 5. Eight test images.

4.1 Analyses of image quality and payload

Table 1 shows the PSNR, payload P_ℓ , and the size of the compressed location map $|cLM|$ of the proposed method for each test images at embedding level $\ell = 1, 5$ and 10, respectively. Here the payload is referred to the pure payload, i.e., the actual amount of embedded secret data. To see how embedding level affects the PSNR and payload, we set threshold $T = 255$ to force every QLs to join the embedding processes. That is, the threshold mechanism describe in Section 3.3 is not used in **Table 1**.

Table 1 shows that when embedding level $\ell = 1$, the PSNR of the proposed methods always higher than 48.13 dB. This is because for the first embedding level, the higher and lower QLs are modified only by one unit at most. Therefore, the MSE between the images decoded from the original and stego BTC codes is one at most, and thus the lower bound PSNR is $10 \log_{10}(255^2 / 1) \approx 48.13 \text{ dB}$. Table 1 also reveals that the size of the compressed location map $|cLM|$ is increased as the embedding level ℓ increased. This is because more pixels are included in the location map. Moreover, the Tiffany image possesses a larger location map because it contains richer saturated pixels (pixel values near 0 and 255) than other test images.

Table 1. Comparison of PNSR and pure payload

Image	$\ell = 1$			$\ell = 5$			$\ell = 10$		
	PSNR	P_ℓ	$ cLM $	PSNR	P_ℓ	$ cLM $	PSNR	P_ℓ	$ cLM $
Lena	48.47	19378	216	35.72	28869	216	30.46	35386	216
Jet	49.32	20224	216	35.60	29143	216	30.12	35497	216
Peppers	49.06	19192	216	36.33	28373	472	30.40	34563	1312
Boat	49.38	18088	216	35.75	24628	216	30.27	29428	488
Tiffany	48.59	19987	544	35.94	28500	1696	29.99	33105	3496
Tank	49.91	18439	216	36.42	26642	216	31.26	33196	216
Truck	49.09	17810	216	35.42	23953	216	30.65	29661	328
Baboon	48.75	16920	256	34.88	19655	392	29.03	22618	336

Figs. 6(a) and (b) show the payload-PSNR curves of the Lena and Baboon images under various thresholds, respectively. As can be seen from the figures, smaller threshold provides a higher image quality but offers a lower payload. A smaller threshold strictly selects only those most likely embeddable differences in the embedding process and thus it produces a higher stego image quality with the sacrifice of the payload. On the contrary, a large threshold selects more differences in the embedding process and thus it provides a higher payload with lower image quality.

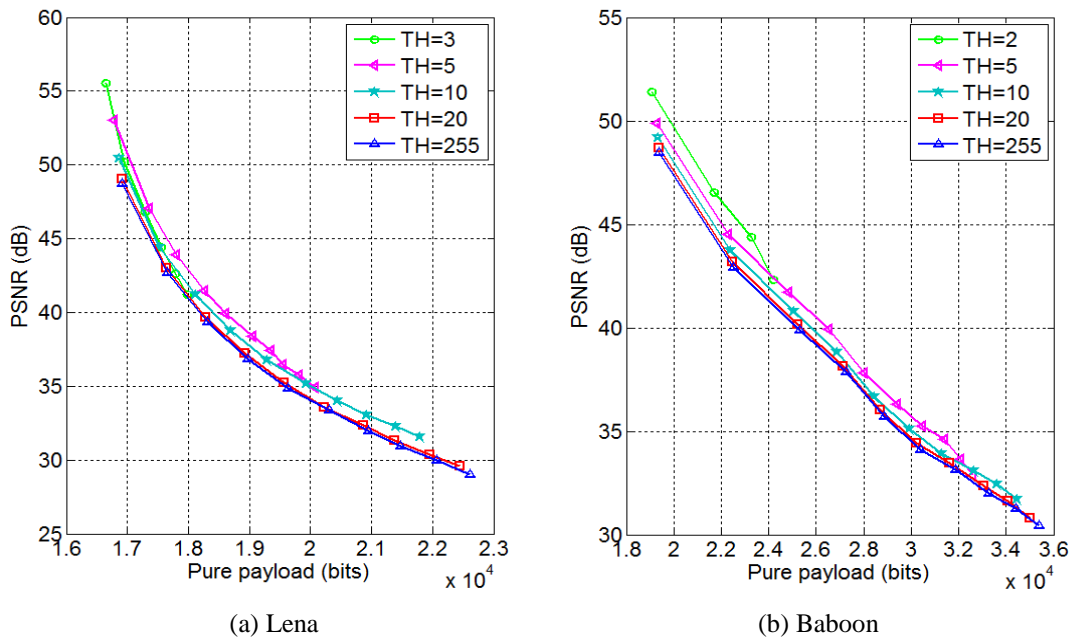


Fig. 6. PSNR versus payload for various thresholds.

Since the selection of threshold affect the payload and image quality, the threshold can be properly selected such that a best image quality is achieved for given a payload and a pre-determined embedding level ℓ . Such a threshold can be determined by using the bisection method with a few iterations. To perform the bisection method, since the payload P_ℓ is a function of the threshold T , we have $P_\ell = f(T)$. Because the smallest payload $P_{\ell,0}$ and the

largest payload $P_{\ell,255}$ occur at $T=0$ and $T=255$, respectively, we have the initial conditions $P_{\ell,0} = f(0)$ and $P_{\ell,255} = f(255)$. Finally, the desirable threshold T for given a payload P_{ℓ} can then be determined by applying the bisection method with the initial conditions $P_{\ell,0} = f(0)$ and $P_{\ell,255} = f(255)$.

Fig. 7 shows a plot of payload-PSNR curves for the eight test images with the threshold mechanism is applied. The results show that the Jet image offers the best embedding performance and the Baboon image performs the worst. This is because the proposed method shifts the QLs with the most frequently occurring QL differences and they are richer in smooth images and thus the payload is higher. On the contrary, the QL differences of complex images such as Baboon tend to distribute in a wider range, and thus the number of the most frequently occurring difference is decreased. Therefore, the Baboon image has lower payload compared to other smoother test images.

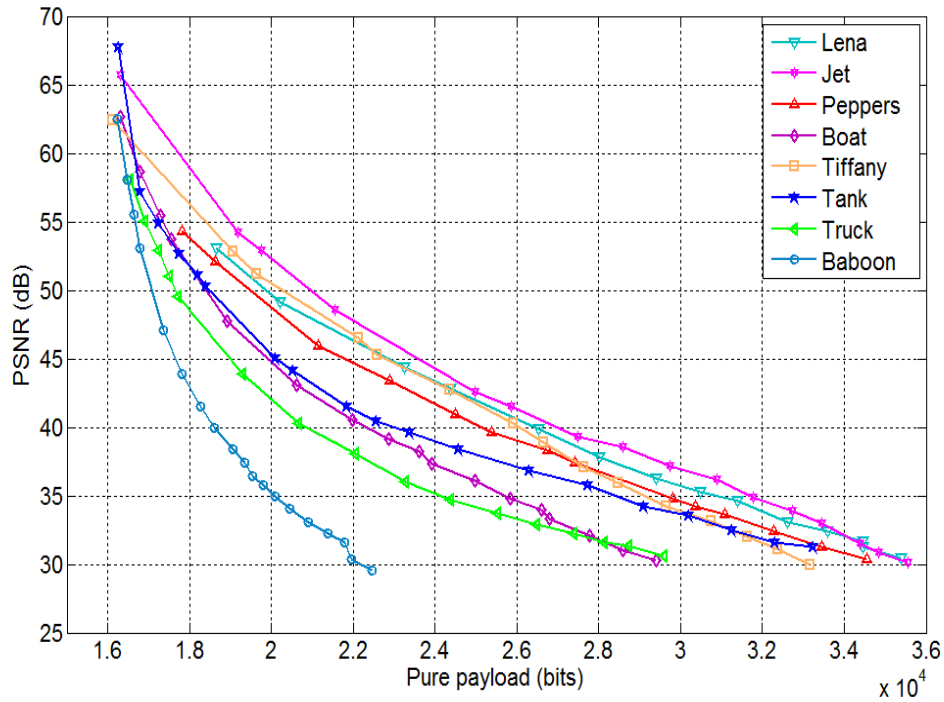


Fig. 7. Performance comparison for eight test images.

4.2 Comparison with Tsai et al.'s method

In this section, we compare the performance of the proposed method with Tsai et al.'s method [25] for four test images Lena, Jet, Peppers, and Baboon. The comparison results are shown in **Figs. 8 (a)-(d)**, respectively. As can be seen from the figures, the proposed method offers significantly better image quality than that of Tsai et al.'s method, especially at low payload. This is because the proposed QL-shifting technique does not change the numerical order of QLs and therefore, QL-swapping technique can be applied before applying QL-shifting technique to obtain a no-distortion version of stego image. However, when the payload is larger than $N - N_{eq}$, QL-shifting has to be applied to acquire more room for data embedding

and thus the distortion inevitably occurs. Nevertheless, the proposed method also offers better image quality than that of Tsai et al.'s method.

The proposed method performs the best when the threshold mechanism is also applied. Although **Figs. 8 (a)-(d)** only shows the comparison results for four test images, other test images including Boat, Tiffany, Tank, and Truck also show the similar results, indicating that the proposed method has better embedding performance than that of Tsai et al.'s method.

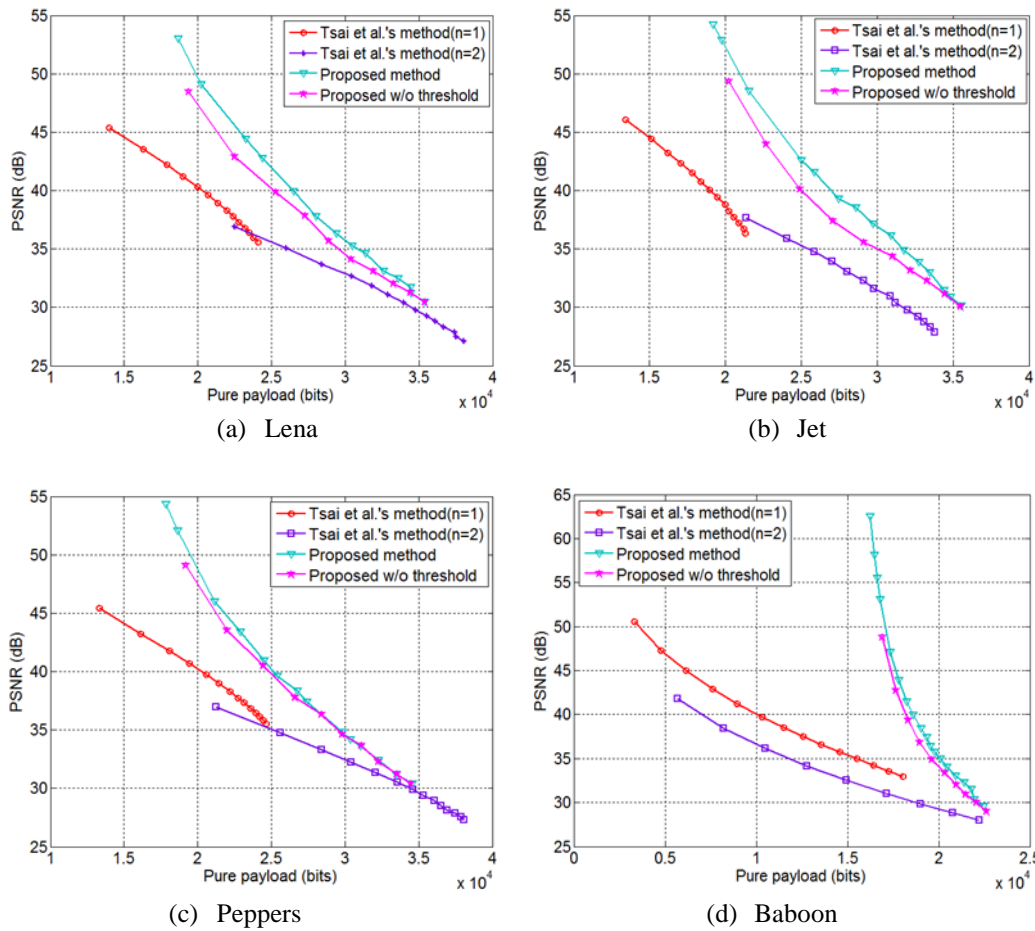


Fig. 8. Performance comparison of the proposed method and Tsai et al.'s method.

4.3 Comparison with other related method

In this section, we compared the proposed method with other related methods, including Chen et al.'s [22], Lo et al.'s [24] and Tsai et al.'s [25] methods under various payloads. In Chen et al.'s method, additional 16 bits are embedded when the higher and lower QLs are equal to achieve the best payload. In Lo et al.'s method, the data set of Choice 2 is used and two pairs of peaks and zeros are employed because it achieves the best embedding performance. In Tsai et al.'s method, the best parameters n and EP are selected such that the best embedding performance can be obtained. The results are shown in **Table 2**. As seen from the table, the proposed method achieves a no-distortion result ($PSNR = \infty$) when the payload is small (4000 and 10000 bits). On the contrary, only Chen et al.'s method achieves no-distortion results but

others give distorted versions. Note that although Lo et al.'s method offers high image quality at low payload, the hiding capacity is limited and none of the images can embed secret data more than 10,000 bits. When the payload is raised to 20,000 and 30,000 bits, no image in Chen et al.'s and Lo et al.'s methods can embed the secret data at this size, and the proposed method offers a significantly better image quality than Tsai et al.'s method. The experimental results indicate that the proposed method offers no-distortion embedding at low payload while achieving a better image quality than other methods at high payload.

Table 2. PSNR comparison of related methods for various payload

		Lena	Jet	Peppers	Boat	Tiffany	Tank	Truck	Baboon
4000 bits	Chen et al.	∞	∞	∞	∞	∞	∞	∞	∞
	Lo et al.	54.27	53.87	52.17	n/a	58.32	47.98	n/a	n/a
	Tsai et al.	n/a	n/a	n/a	54.62	n/a	53.24	53.60	49.03
	Proposed	∞	∞	∞	∞	∞	∞	∞	∞
10000 bits	Chen et al.	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Lo et al.	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Tsai et al.	48.86	49.44	48.22	46.55	49.10	45.05	44.76	40.07
	Proposed	∞	∞	∞	∞	∞	∞	∞	∞
20000 bits	Chen et al.	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Lo et al.	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Tsai et al.	40.33	38.75	40.24	37.20	40.71	37.60	36.10	29.25
	Proposed	49.68	52.32	48.74	44.77	50.54	45.32	42.03	35.19
30000 bits	Chen et al.	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Lo et al.	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Tsai et al.	32.89	31.47	32.46	29.67	32.84	29.75	28.30	n/a
	Proposed	35.73	36.91	34.62	29.59	33.88	33.67	30.25	20.13

5. Conclusions

In this paper, we proposed a reversible data hiding method for BTC compressed images. Three embedding techniques are utilized to enhance the stego image quality. The QL-swapping technique produces a no-distortion version of stego codes by swapping the QLs. QL-shifting technique shifts the differences between the two QLs for data embedding, and the threshold mechanism can also be applied to further enhance the image quality. The experimental results revealed that the proposed method not only provides a larger payload and better image quality than prior works, but also offers a no-distortion version of stego codes at low payload.

References

- [1] Z. Fu, X. Sun, Q. Liu, L. Zhou, and J. Shu, "Achieving Efficient Cloud Search Services: Multi-keyword Ranked Search over Encrypted Cloud Data Supporting Parallel Computing," *IEICE Transactions on Communications*, vol. E98-B, no. 1, pp.190-200, 2015. [Article \(CrossRef Link\)](#)
- [2] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A Secure and Dynamic Multi-keyword Ranked Search Scheme over Encrypted Cloud Data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340-352, 2016. [Article \(CrossRef Link\)](#)
- [3] Y. Ren, J. Shen, J. Wang, J. Han, and S. Lee, "Mutual Verifiable Provable Data Auditing in Public Cloud Storage," *Journal of Internet Technology*, vol. 16, no. 2, pp. 317-323, 2015. [Article \(CrossRef Link\)](#)

- [4] P. Guo, J. Wang, B. Li, and S. Lee, "A Variable Threshold-value Authentication Architecture for Wireless Mesh Networks," *Journal of Internet Technology*, vol. 15, no. 6, pp. 929-936, 2014. [Article \(CrossRef Link\)](#)
- [5] J. Li, X. Li, B. Yang and X. Sun, "Segmentation-based Image Copy-move Forgery Detection Scheme," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 507-518, 2015. [Article \(CrossRef Link\)](#)
- [6] B. Chen, H. Shu, G. Coatrieux, G. Chen, X. Sun, and J.L. Coatrieux, "Color Image Analysis by Quaternion-Type Moments," *Journal of Mathematical Imaging and Vision*, vol. 51, no. 1, pp. 124-144, 2015. [Article \(CrossRef Link\)](#)
- [7] Z. Xia, X. Wang, X. Sun, Q. Liu, and N. Xiong, "Steganalysis of LSB Matching Using Differences Between Nonadjacent Pixels," *Multimedia Tools and Applications*, vol. 75, no. 4, pp. 1947-1962, 2016. [Article \(CrossRef Link\)](#)
- [8] Z. Xia, X. Wang, X. Sun, and B. Wang, "Steganalysis of Least Significant Bit Matching Using Multi-Order Differences," *Security and Communication Networks*, vol. 7, no. 8, pp. 1283-1291, 2014. [Article \(CrossRef Link\)](#)
- [9] B. Gu, V.S. Sheng, K.Y. Tay, W. Romano, and S. Li, "Incremental Support Vector Learning for Ordinal Regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 7, pp. 1403-1416, 2015. [Article \(CrossRef Link\)](#)
- [10] B. Gu, V. S. Sheng, Z. Wang, D. Ho, S. Osman, and S. Li, "Incremental Learning for v-Support Vector Regression," *Neural Networks*, vol. 67, pp. 140-150, 2015. [Article \(CrossRef Link\)](#)
- [11] X. Wen, L. Shao, Y. Xue, and W. Fang, "A Rapid Learning Algorithm for Vehicle Classification," *Information Sciences*, vol. 295, no. 1, pp. 395-406, 2015. [Article \(CrossRef Link\)](#)
- [12] Z. Yuhui, J. Byeungwoo, X. Danhua, W.Q.M. Jonathan and Z. Hui, "Image Segmentation by Generalized Hierarchical Fuzzy C-means Algorithm," *Journal of Intelligent and Fuzzy Systems*, vol. 28, no. 2, pp. 961-973, 2015. [Article \(CrossRef Link\)](#)
- [13] J. Mielikainen, "LSB Matching Revisited," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 285-287, 2006. [Article \(CrossRef Link\)](#)
- [14] W. Hong and T.S. Chen, "A Novel Data Embedding Method Using Adaptive Pixel Pair Matching," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 176-184, 2012. [Article \(CrossRef Link\)](#)
- [15] Z. Ni, Y.Q. Shi, N. Ansari, and W. Su, "Reversible Data Hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354-362, 2006. [Article \(CrossRef Link\)](#)
- [16] X.L. Li, J. Li, B. Li, and B. Yang, "High-Fidelity Reversible Data Hiding Scheme Based on Pixel-Value-Ordering and Prediction-Error Expansion," *Signal Processing*, vol.93, no. 1, pp. 198-205, 2013. [Article \(CrossRef Link\)](#)
- [17] X. Wang, J. Ding, and Q.Q. Pei, "A Novel Reversible Image Data Hiding Scheme Based on Pixel Value Ordering and Dynamic Pixel Block Partition," *Information Science*, vol. 310, pp. 16-35, 2015. [Article \(CrossRef Link\)](#)
- [18] W. Hong, T.S. Chen, and J. Chen, "Reversible Data Hiding Using Delaunay Triangulation and Selective Embedment," *Information Sciences*, vol. 308, pp. 140-154, 2015. [Article \(CrossRef Link\)](#)
- [19] W.J. Wang, C.T. Huang, C.M. Liu, P.C. Su, and S.J. Wang, "Data Embedding for Vector Quantization Image Processing on the Basis of Adjoining State-Codebook Mapping," *Information Sciences*, vol. 246, pp. 69-82, 2013. [Article \(CrossRef Link\)](#)
- [20] X. Zhang, S. Wang, Z. Qian, and G. Feng, "Reversible Fragile Watermarking for Locating Tampered Blocks in JPEG Images," *Signal Processing*, vol. 90, no. 12, pp. 3026-3036, 2010. [Article \(CrossRef Link\)](#)
- [21] W. Hong, T.S. Chen, C.W. Shiu, and M.C. Wu, "Lossless Data Embedding in BTC Codes Based on Prediction and Histogram Shifting," *Applied Mechanics and Materials*, vol. 65, pp. 182-185, 2011. [Article \(CrossRef Link\)](#)
- [22] J. Chen, W. Hong, T.S. Chen and C.W. Shiu, "Steganography for BTC Compressed Images Using No Distortion Technique," *Image Science Journal*, vol. 58, no. 4, pp. 177-185. 2010. [Article \(CrossRef Link\)](#)

- [23] C.C. Chang, C.Y. Lin, and Y.H. Fan, "Reversible Steganography for BTC-Compressed Images," *Fundamenta Informaticae*, vol. 109, no. 2, pp. 121-134, 2011. [Article \(CrossRef Link\)](#)
- [24] C.C. Lo, Y.C. Hu, W.L. Chen, and C.M. Wu, "Reversible Data Hiding Scheme for BTC-Compressed Images Based on Histogram Shifting," *International Journal of Security and Its Applications*, vol. 8, no. 2, pp. 301-314, 2014. [Article \(CrossRef Link\)](#)
- [25] Y.Y. Tsai, C.S. Chan, C.L. Liu, and B.R. Su, "A Reversible Steganographic Algorithm for BTC-Compressed Images Based on Difference Expansion and Median Edge Detector," *Imaging Science Journal*, vol. 62, no. 1, pp. 48-55, 2014. [Article \(CrossRef Link\)](#)
- [26] M.D. Lema and O.R. Mitchell, "Absolute Moment Block Truncation Coding and Its Application to Color Images," *IEEE Transactions on Communication*, vol. 32, no. 10, pp. 1148-1157, 1984. [Article \(CrossRef Link\)](#)
- [27] The USC-SIPI Image Database [Online]. Available: <http://sipi.usc.edu/database/>



Wien Hong received his M.S. and Ph.D. degree from the State University of New York at Buffalo, USA in 1994 and 1997, respectively. He is currently a researcher at Nanjing University of Information Science & Technology and Nanfang College of Sun Yat-Sen University. His research interests include steganography, watermarking and image compression.



Shuozen Zheng is with the Department of Electronic Communication and Software Engineering, Nanfang College of Sun Yat-Sen University since 2014. His research interests include data hiding and communication technology.



Tung-Shou Chen received his B.S. and Ph.D. degrees from National Chiao Tung University in 1986 and 1992, respectively, both in Computer Science and Information Engineering. Currently, he is a professor and head of the Department of Computer Science and Information Engineering at National Taichung University of Science and Technology, Taichung, Taiwan. His current research interests include data mining, image cryptosystems, steganography, image compression and bioinformatics.



Chien-Che Huang received his Ph.D. degrees from National Changhua University of Education, and since 2014, he is an assistant professor at Department of Computer Science and Information Engineering at National Taichung University of Science and Technology, Taiwan. His research interests include data hiding, data mining, and cloud computing.