Reversible Data Hiding Exploiting Variance in Wavelet Coefficients

Xu-Ren Luo
Department of Electrical and Electronic Engineering, Chung Cheng Institute of Technology,
National Defense University,
Tahsi, Taoyuan 33509, Taiwan, Republic of China.
and
Te-Lung Yin
Department of Computer Science and Information Engineering, China University of Technology,
Hukou, Hsinchu 303, Taiwan, Republic of China.

ABSTRACT

In this paper, we present a new reversible data hiding scheme that utilizes the wavelet transform and better exploits the large wavelet coefficient variance to achieve high capacity and imperceptible embedding. Our scheme differs from those of previous studies in that the wavelet coefficients histogram rather than the gray-level histogram is manipulated. In addition, we design intelligent histogram-shifting rules to avoid the decimal problem in grayscale pixel values after recovery process to achieve reversibility. Small changes in the wavelet coefficients after embedding process are important factors contributing to low visual distortion in the marked image. Furthermore, an important property of our scheme is that the use of threshold differs greatly from previous schemes. The experimental results show that our scheme outperforms other reversible data hiding schemes.

Keywords: reversibility, marked media, wavelet transform, wavelet coefficient, distortion, histogram.

1. Introduction

Reversible data hiding, or so-called invertible, distortion-free data hiding, is a branch of fragile technique mainly used for quality-sensitive applications such as multimedia content authentication, medical imaging systems, law enforcement, and military imagery, etc. One of the most important requirements in these fields is to recover the original media exactly during analysis to enable the right decisions. The other significant necessities of reversible data hiding are the embedding capacity and visual quality of the marked media, since they are critically essential to achieving satisfactory performance in various applications. [1][2]

The scheme we present in this paper is an attempt to achieve high-performance reversible data hiding, in which the embedding and recovering processes are devised in the frequency domain. The particularities of large variance in wavelet coefficients and minor changes in wavelet coefficients following from the embedding process in wavelet coefficients are exploited to achieve high capacity and imperceptibility. The rest of this paper is organized as follows. In Section 2, previous reversible data hiding schemes and their characteristics will be briefly reviewed in terms of embedding capacity and visual quality. Our proposed scheme is introduced in Section 3. Experimental results and comparative analyses are presented in Section 4. Finally, some conclusions are drawn in Section 5.

2. Related studies

Nowadays, a number of research works in this field can be classified into two major categories according to the embedding strategies. Category-I reversible data hiding schemes work on the transform domain. In 2002, Fridrich *et al*. [3][4] proposed a novel paradigm of lossless data embedding. The payload of this so-called RS scheme is highly dependent on the compression algorithm and insufficient for some applications. Subsequently, Xuan *et al*. [5] presented a new scheme carried out in the integer wavelet transform (IWT) domain. In this scheme, one or multiple middle bit-plan(s) in the high-frequency sub-bands is(are) chosen to embed data bits. In 2003, Tian [6] proposed a difference expansion (DE) scheme, which exploited the DE technique to embed data bits into the high-frequency coefficients. However, this scheme suffers from the location map problem that it is difficult to achieve capacity control. Alattar [7][8] extended Tian's scheme by generalizing the DE technique to the triplets and quads of adjacent pixels. Kamstra *et al*. [9][10] improved the DE scheme by predicting the expandable locations in the high-pass band. This scheme improves the efficiency of lossless compression, although the embedding capacity is small. In 2007, Thodi *et al*. [11] proposed a new scheme combining histogram shifting and prediction-error expansion approaches to remedy the problems of Tian's scheme.

In category-II, schemes are performed in the spatial domain. Fridrich *et al.* [12] presented the joint bi-level image experts group (JBIG) lossless compression technique to save space for data embedding. However, the payload is highly dependent on the lossless compression algorithm. Celik *et al.* [13][14] employed the generalized least-significant-bit (G-LSB) technique and the context-based adaptive lossless image coding (CALIC) to achieve lossless data hiding. In 2006, Ni *et al.* [15] utilized the pairs of maximum and minimum points of a given image histogram to achieve reversibility. In this scheme, the pixels between each pair are modified for data embedding and extraction. However, such a strategy may lead to significant overhead and insufficient visual quality. In [16][17], Hwang *et al.* and Kuo *et al.* extend Ni's scheme by using a location map to achieve reversibility. Although the scheme is simple, the significant overhead and insufficient visual quality are critical problems. In 2009, Kim *et al.* [18] exploited the feature of high spatial correlation between neighboring pixels to achieve high-performance data hiding. The embedding capacity in the paper ranges from 6 to 210 k.

For all of the above reversible data hiding schemes, the requirement of additional overhead is one of the thorniest problems in the restore process. This paper presents a novel method and enhances the embedding performance recently proposed by Luo *et al*. [19] in the frequency domain to achieve high-performance lossless data hiding.

3. Proposed scheme

The proposed scheme combines the two-level Haar discrete wavelet transform (HDWT) algorithm and a new histogram shifting technique to achieve reversible data hiding. In our scheme, a given image is first transformed into a frequency domain and sub-bands in the middle- and high-frequency ranges are then used to create sub-band differences. Each histogram of these sub-band differences is then shifted according to a selected threshold. Message bits can then be embedded in the empty space of the shifted histograms. Finally, the marked image is reconstructed with the sub-bands carrying and non-carrying hidden message by performing the inverse HDWT algorithm to complete the embedding process. As to the extracting process, the corresponding inverse operations can be performed to recover the hidden information and the original image. The new scheme differs from our previous study [19] in that six rather than four of the wavelet sub-bands are manipulated.

3.1. Segmentation algorithm

The two-level HDWT algorithm utilizes the four-band sub-band coding system to decompose an image into a set of different frequency sub-bands. As illustrated in Figs. 1 and 2, the size of each sub-band is one eighth of the original image in the spatial domain. The eight different sub-bands can be classified into the low-, middle-, and high-frequency sub-bands. Since the low-frequency sub-band of an image incorporates more energy than the other sub-bands, its coefficients are the most fragile that if any of them are manipulated, a suspect can visibly detect the changes on the spatial domain image. In contrast, if the coefficients in the middle- and/or high-frequency sub-bands are altered, changes in the spatial domain image are imperceptible to human eyes. As a result, this feature is generally exploited to conceal secret messages.

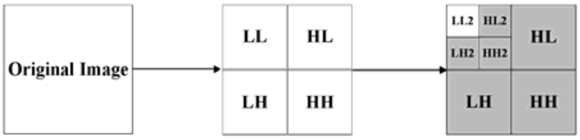


Fig. 1. The two-level segmentation process.

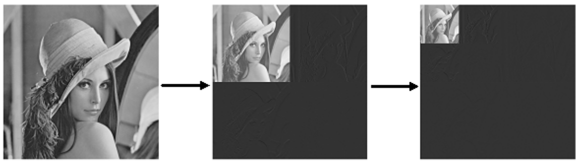


Fig. 2. A tow-level HDWT four-band split of “Lena”.

3.2. Data embedding algorithm

We assume that the embedded message is a random binary sequence. The histograms of the sub-band differences between the reference sub-band and the other destination sub-bands are shifted to embed the secret message. Fig. 3 depicts the overall data embedding process, which is described in detail below.

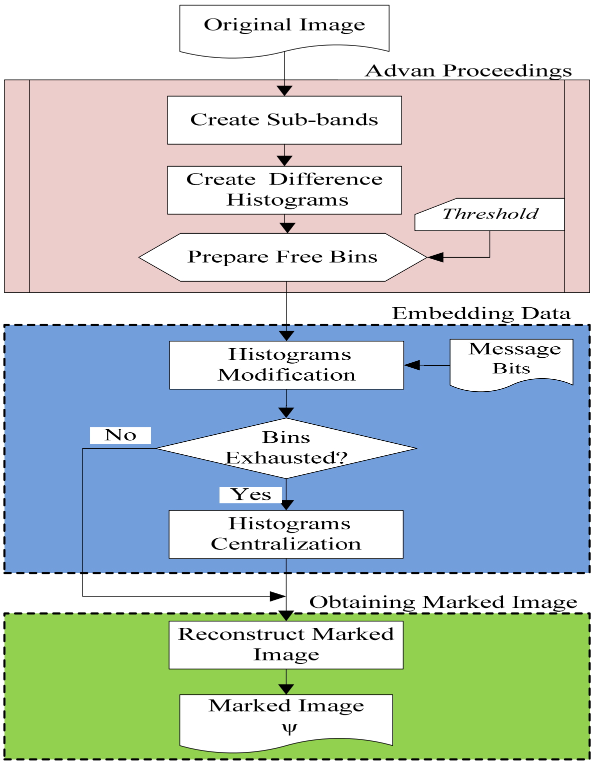


Fig. 3. Flowchart of data embedding process.

Embedding_process (η, T, φ)

Input: η, the original image; *T*, the threshold according to which the empty bins in each histogram are prepared; φ, the secret message.

Output: ψ, the marked image, f, the mark of embedding status.

Advan Proceedings:

Step 1: Create sub-bands by performing two-level HDWT four-band sub-band coding system on an input image η. Six of the sub-bands to be utilized are denoted by $LH(x,y)$, $HL(x,y)$, $HH(x,y)$, $LH2(x,y)$, $HH2(x,y)$ and $HL2(x,y)$, where (x,y) indicates the coordinate of the coefficients in each sub-band.

Step 2: Create sub-band differences D_1, D_2, D_3 and D_4 between the reference sub-bands LH, $LH2$ and the other destination sub-bands HL, HH, $HL2$ and $HH2$ by the following formulas:

$$D_1(x,y) = LH(x,y) - HL(x,y), \quad (1)$$

$$D_2(x,y) = LH(x,y) - HH(x,y), \quad (2)$$

$$D_3(x,y) = LH2(x,y) - HL2(x,y), \quad (3)$$

$$D_4(x,y) = LH2(x,y) - HH2(x,y). \quad (4)$$

Step 3: Denote the histograms of D_i as h_i, where

$i = 1, 2, 3$ and 4.

Step 4: Shift histogram h_i according to the threshold T selected. The shifted h_i can be calculated as follows:

$$h_i' = \begin{cases} h_i(j) + 8, & \text{if } h_i(j) \geq T + 1, \\ h_i(j) - 8, & \text{if } h_i(j) \leq -(T+1), \end{cases} \tag{5}$$

where $i = 1, 2, 3, 4$ and (j) indicates the value of each bin. These can also be obtained by the following formulas:

$$D_1'(x, y) = LH(x, y) - HL'(x, y), \tag{6}$$

$$D_2'(x, y) = LH(x, y) - HH'(x, y), \tag{7}$$

$$D_3'(x, y) = LH2(x, y) - HL2'(x, y), \tag{8}$$

$$D_4'(x, y) = LH2(x, y) - HH2'(x, y). \tag{9}$$

where $HL'(x,y)$, $HH'(x,y)$, $HL_2'(x,y)$ and $HH_2'(x,y)$ can be expressed as:

$$HL'(x,y) = \begin{cases} HL(x,y) - 8, & \text{if } h_1(j) \geq T + 1, \\ HL(x,y) + 8, & \text{if } h_1(j) \leq -(T+1). \end{cases} \tag{10}$$

$$HH'(x,y) = \begin{cases} HH(x,y) - 8, & \text{if } h_2(j) \geq T + 1, \\ HH(x,y) + 8, & \text{if } h_2(j) \leq -(T+1). \end{cases} \tag{11}$$

$$HL2'(x,y) = \begin{cases} HL2(x,y) - 8, & \text{if } h_3(j) \geq T + 1, \\ HL2(x,y) + 8, & \text{if } h_3(j) \leq -(T+1). \end{cases} \tag{12}$$

$$HH2'(x,y) = \begin{cases} HH2(x,y) - 8, & \text{if } h_4(j) \geq T + 1, \\ HH2(x,y) + 8, & \text{if } h_4(j) \leq -(T+1). \end{cases} \tag{13}$$

Embedding Data:

Step 5: We first set an interation index ℓ to T and then embed message bits sequentially by modifying h_i'. Each h_i' is shifted to become h_i'', where $i = 1, 2, 3$ and 4 by the following rules:

$$h_i'' = \begin{cases} h_i'(j) + 8, & \text{if } h_i'(j) = \ell, \ \varphi(n) = 0, \\ h_i'(j) + 4, & \text{if } h_i'(j) = \ell, \ \varphi(n) = 1, \\ h_i'(j) - 8, & \text{if } h_i'(j) = -\ell, \ \varphi(n) = 0, \\ h_i'(j) - 4, & \text{if } h_i'(j) = -\ell, \ \varphi(n) = 1, \end{cases} \tag{14}$$

for $\ell > 0$.

$$h_i'' = \begin{cases} h_i'(j) + 8, & \text{if } h_i'(j) = 0, \ \phi(n) = 0, \\ h_i'(j) + 4, & \text{if } h_i'(j) = 0, \ \phi(n) = 1, \end{cases} \tag{15}$$

for $\ell = 0$.

The changes in the difference histograms above result in changes in the coefficients. This implies that $D_i'(x, y)$ is scanned and modified again. Once the value of $D_i'(x, y)$ is equal to $\pm\ell$, the message bit is embedded. This process is repeated until there are no $D_i'(x, y)$ with the value of $\pm\ell$. We then decrease ℓ by 1 and repeat the step until $\ell < 0$. These steps can be formulated as follows:

$$D_1''(x, y) = LH(x, y) - HL''(x, y), \tag{16}$$

$$D_2''(x, y) = LH(x, y) - HH''(x, y), \tag{17}$$

$$D_3''(x, y) = LH2(x, y) - HL2''(x, y), \tag{18}$$

$$D_4''(x, y) = LH2(x, y) - HH2''(x, y). \tag{19}$$

if $\ell > 0$,

$$HL''(x,y) = \begin{cases} HL'(x,y) + 4, & \text{if } h_1'(j) = -\ell, \ \phi(n) = 1, \\ HL'(x,y) - 4, & \text{if } h_1'(j) = \ell, \ \phi(n) = 1, \\ HL'(x,y) + 8, & \text{if } h_1'(j) = -\ell, \ \phi(n) = 0, \\ HL'(x,y) - 8, & \text{if } h_1'(j) = \ell, \ \phi(n) = 0. \end{cases} \tag{20}$$

$$HH''(x,y) = \begin{cases} HH'(x,y) + 4, & \text{if } h_2'(j) = -\ell, \ \phi(n) = 1, \\ HH'(x,y) - 4, & \text{if } h_2'(j) = \ell, \ \phi(n) = 1, \\ HH'(x,y) + 8, & \text{if } h_2'(j) = -\ell, \ \phi(n) = 0, \\ HH'(x,y) - 8, & \text{if } h_2'(j) = \ell, \ \phi(n) = 0. \end{cases} \tag{21}$$

$$HL2''(x,y) = \begin{cases} HL2'(x,y) + 4, & \text{if } h_3'(j) = -\ell, \ \phi(n) = 1, \\ HL2'(x,y) - 4, & \text{if } h_3'(j) = \ell, \ \phi(n) = 1, \\ HL2'(x,y) + 8, & \text{if } h_3'(j) = -\ell, \ \phi(n) = 0, \\ HL2'(x,y) - 8, & \text{if } h_3'(j) = \ell, \ \phi(n) = 0. \end{cases} \tag{22}$$

$$HH2''(x,y) = \begin{cases} HH2'(x,y) + 4, & \text{if } h_4'(j) = -\ell, \ \phi(n) = 1, \\ HH2'(x,y) - 4, & \text{if } h_4'(j) = \ell, \ \phi(n) = 1, \\ HH2'(x,y) + 8, & \text{if } h_4'(j) = -\ell, \ \phi(n) = 0, \\ HH2'(x,y) - 8, & \text{if } h_4'(j) = \ell, \ \phi(n) = 0. \end{cases} \tag{23}$$

if $\ell = 0$,

$$HL''(x,y) = \begin{cases} HL'(x,y) - 4, & \text{if } h_1'(j) = 0, \ \phi(\text{n}) = 1, \\ HL'(x,y) - 8, & \text{if } h_1'(j) = 0, \ \phi(\text{n}) = 0. \end{cases} \tag{24}$$

$$HH''(x,y) = \begin{cases} HH'(x,y) - 4, & \text{if } h_2'(j) = 0, \ \phi(\text{n}) = 1, \\ HH'(x,y) - 8, & \text{if } h_2'(j) = 0, \ \phi(\text{n}) = 0. \end{cases} \tag{25}$$

$$HL2''(x,y) = \begin{cases} HL2'(x,y) - 4, & \text{if } h_3'(j) = 0, \ \phi(\text{n}) = 1, \\ HL2'(x,y) - 8, & \text{if } h_3'(j) = 0, \ \phi(\text{n}) = 0. \end{cases} \tag{26}$$

$$HH2''(x,y) = \begin{cases} HH2'(x,y) - 4, & \text{if } h_4'(j) = 0, \ \phi(\text{n}) = 1, \\ HH2'(x,y) - 8, & \text{if } h_4'(j) = 0, \ \phi(\text{n}) = 0. \end{cases} \tag{27}$$

Centralizing Histogram

Step 6: When all the bins in the difference histogram are exhausted, eight bins, valued from -4 to 3, will become empty. In this case, the mark f is set to be "1" and all bins on the right side will be moved left 4 bins, and those on the left will be moved right 4 bins in order to improve the visual quality by decreasing the variance of the differences in the coefficients. Otherwise, the mark f is set to be "0". Each h_i'' is shifted to become h_i''' by the following rules:

$$h_i'''(j) = \begin{cases} h_i''(j) - 4, & \text{if } h_i''(j) > 0, \\ h_i''(j) + 4, & \text{if } h_i''(j) < 0, \end{cases} \tag{28}$$

where $i = 1, 2, 3$ and 4.

Shifting histograms as above creates coefficient changes, which can be formulated as follows:

$$D_1'''(x, y) = LH(x, y) - HL'''(x, y), \tag{29}$$

$$D_2'''(x, y) = LH(x, y) - HH'''(x, y), \tag{30}$$

$$D_3'''(x, y) = LH2(x, y) - HL2'''(x, y)\ , \tag{31}$$

$$D_4'''(x, y) = LH2(x, y) - HH2'''(x, y)\ , \tag{32}$$

where $HL'''(x,y)$, $HH'''(x,y)$, $HL_2'''(x,y)$ and $HH_2'''(x,y)$ can be expressed as:

$$HL'''(x, y) = \begin{cases} HL''(x,y)+4, & \text{if } h_1''(j) > 0, \\ HL''(x,y)-4, & \text{if } h_1''(j) < 0. \end{cases} \tag{33}$$

$$HH'''(x, y) = \begin{cases} HH''(x,y)+4, & \text{if } h_2''(j) > 0, \\ HH''(x,y)-4, & \text{if } h_2''(j) < 0. \end{cases} \tag{34}$$

$$HL2'''(x, y) = \begin{cases} HL2''(x,y)+4, & \text{if } h_3''(j) > 0, \\ HL2''(x,y)-4, & \text{if } h_3''(j) < 0. \end{cases} \tag{35}$$

$$HH2'''(x, y) = \begin{cases} HH2''(x,y)+4, & \text{if } h_4''(j) > 0, \\ HH2''(x,y)-4, & \text{if } h_4''(j) < 0. \end{cases} \tag{36}$$

Step 7: Using the sub-bands *HL* and *HL2* with the hidden message as the reference sub-bands, and then fifth and sixth difference histogram is created with the original sub-bands *LH* and *LH2*. The secret message is then hidden into these difference histograms according to steps 1 to 6; as a result, $LH'''(x, y)$ and $LH2'''(x, y)$ will be created. This completes the embedding process.

Obtaining Marked Image:

Step 8: Reconstruct the marked image by utilizing the inverse of the two-level HDWT four-band sub-band coding on these sub-bands. We first reconstruct the *LL* sub-band and then reconstruct the marked image ψ. This procedure can be formulated as follows:

$$LL = IDWT(LL2, LH2''', HL2''', HH2''')\ , \tag{37}$$

$$\psi = IDWT\ (LL\ , LH''', HL''', HH''')\ . \tag{38}$$

Prevention of overlap and over/underflow

A flag-bit is used to indicate whether the bin in the different histograms is overlap or not. The flag-bit is set to 1 if a bin value shifted by ± 8 overlaps with one shifted by ± 4. The flag-bit is set to 0 if no overlap occurs. These flag-bits and the values of *T* and *f* in the bitmap will ultimately all be compressed with an efficient compression tool based on the LZMA algorithm. The compressed result is then hidden into the reserved non-overlapping bins. According to experimental results, this process results in less than 2.92% overhead in entire embedding capacity for many different types of images.

Though the generated pixel values in the marked image may be outside the allowable range, the method in [12][18] could be used to deal with the problem. The interval range is dynamically selected with the image characteristics that it will reduce distortions to minimum.

3.3 Data extracting and recovering algorithm

Before extracting the hidden message, receiver needs to verify whether or not the marked image has been modified. If there is more than one occurrence at $h_{\psi i}(j) = -1$, we can conclude that the marked image has been tampered with. The proposed scheme then stops the following extraction steps immediately. The extraction and recovery process is schematized in Fig. 4.

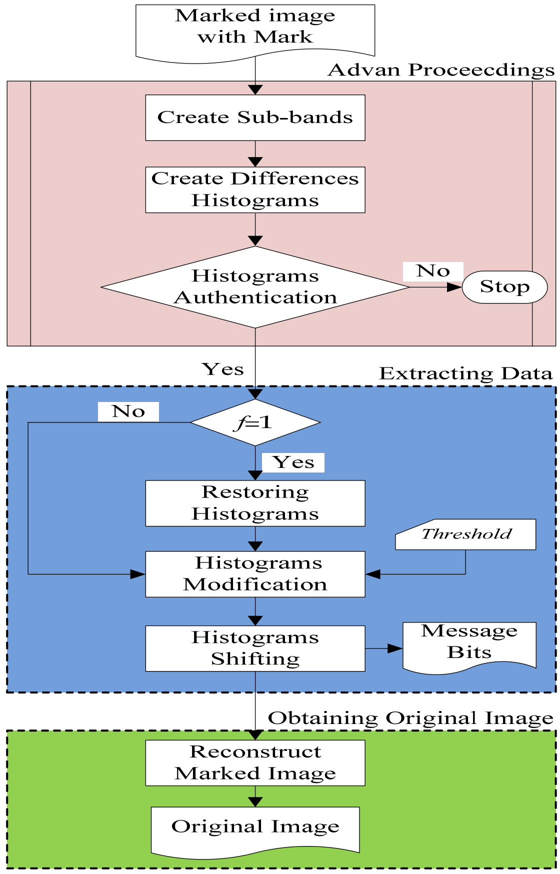


Fig. 4. Flowchart of extraction and recovery process.

The detailed extraction and recovery process includes the following steps:

Extracting_process(ψ, T, f)

Input: ψ, the marked image; *T*, the threshold; f, the mark, according to which the embedding status is determined.

Output: φ, the secret message; η, the original image.

Advan Proceedings:

Step 1: Create sub-bands by performing two-level HDWT four-band coding on the marked image ψ. Six of the sub-bands to be utilized are denoted by $\psi_{LH}(x, y)$, $\psi_{HL}(x, y)$, $\psi_{HL}(x, y)$, $\psi_{LH2}(x, y)$, $\psi_{HL2}(x, y)$ and $\psi_{HH2}(x, y)$, where (x, y) indicates the coordinate of the coefficients in each sub-band.

Recovering *LL* Sub-band

Step 2: Create sub-band difference $D_{\psi 1}(x,y)$ between the reference sub-band ψ_{HL2} and the destination sub-band ψ_{LH2} according to the following formulas:

$$D_{\psi 1}(x, y) = \psi_{HL2}(x, y) - \psi_{LH2}(x, y)\ . \tag{39}$$

Step 3: Denote the histogram of $D_{\psi 1}(x, y)$ as $h_{\psi 1}(j)$, where (j) indicates the value of each bin.

Step 4: Check the distribution of the histogram $h_{\psi 1}(j)$. If there is more than one occurrence at $h_{\psi 1}(j) = -1$, the subsequent steps will be stopped immediately.

Step 5: Check the embedding status. Once the value of *f* is equal to 1, it can be concluded that the sub-band is completely filled with hidden message bits. Subsequently, restore the original difference histogram. The bins greater than or equal to zero will be shifted to the right by 4 and those less than zero to the left by 4. The restored $h'_{\psi 1}(j)$ can be calculated as follows:

$$h'_{\psi 1}(j)=\begin{cases} h_{\psi 1}(j)+4, & \text{if } h_{\psi 1}(j)\geq 0, \\ h_{\psi 1}(j)-4, & \text{if } h_{\psi 1}(j)<0. \end{cases} \tag{40}$$

These can also be obtained by the following formulas:

$$D'_{\psi 1}(x,y)=\psi_{HL2}(x,y)-\psi'_{LH2}(x,y), \tag{41}$$

where $\psi'_{LH2}(x,y)$ can be expressed as:

$$\psi'_{LH2}(x,y)=\begin{cases} \psi_{LH2}(x,y)-4, & \text{if } h_{\psi 1}(j)\geq 0, \\ \psi_{LH2}(x,y)+4, & \text{if } h_{\psi 1}(j)<0. \end{cases} \tag{42}$$

Extracting Data:

Step 6: Extract the hidden message $\varphi(n)$, where *n* denotes the index of a message bit, by shifting $h'_{\psi 1}(j)$ with reference to the bitmap and inverting the embedding process. First, the iteration index ℓ is set to 0. Once a $h'_{\psi 1}(j)$ with a value of $\pm(\ell+4)$ is encountered, a binary bit "1" is retrieved. On the other hand, a binary bit "0" is retrieved if $h'_{\psi 1}(j)$ has a value of $\pm(\ell+8)$. This procedure is repeated until there are no $h'_{\psi 1}(j)$ values of $\pm(\ell+4)$ or $\pm(\ell+8)$. Subsequently, ℓ is increased by 1. The same procedures as described above are repeated until ℓ reaches *T*+1. The retrieving rule is as follows:

$$h''_{\psi 1}=\begin{cases} h'_{\psi 1}(j)-8, & \text{if } h'_{\psi 1}(j)=8, \\ h'_{\psi 1}(j)-4, & \text{if } h'_{\psi 1}(j)=4, \end{cases} \tag{43}$$

for $\ell=0$.

$$h''_{\psi 1}(j)=\begin{cases} h'_{\psi 1}(j)-8, & \text{if } h'_{\psi 1}(j)=\ell+8, \\ h'_{\psi 1}(j)-4, & \text{if } h'_{\psi 1}(j)=\ell+4, \\ h'_{\psi 1}(j)+8, & \text{if } h'_{\psi 1}(j)=-(\ell+8), \\ h'_{\psi 1}(j)+4, & \text{if } h'_{\psi 1}(j)=-(\ell+4), \end{cases} \tag{44}$$

for $1\leq \ell\leq T$.

Step 7: At the same time, the modified sub-band difference $D'_{\psi 1}(x,y)$ is also scanned and modified. The extracting operation can be expressed as the following formula:

$$\phi(n)=\begin{cases} 0, & \text{if } D'_{\psi 1}(x,y)=\pm(\ell+8), \\ 1, & \text{if } D'_{\psi 1}(x,y)=\pm(\ell+4). \end{cases} \tag{45}$$

This procedure is executed until ℓ = *T*+1.

Step 8: Remove the hidden message bits $\phi(n)\in\{0,1\}$ from the sub-band difference. The removing rule is given by

$$D''_{\psi 1}(x,y)=\psi_{HL2}(x,y)-\psi''_{LH2}(x,y), \tag{46}$$

where $\psi''_{LH2}(x,y)$ can be expressed as

$$\psi''_{LH2}(x,y)=\begin{cases} \psi'_{LH2}(x,y)+8, & \text{if } h'_{\psi 1}(j)=8,\ \varphi(n)=0, \\ \psi'_{LH2}(x,y)+4, & \text{if } h'_{\psi 1}(j)=4,\ \varphi(n)=1, \end{cases} \tag{47}$$

for $\ell=0$, and

$$\psi''_{LH2}(x,y)=\begin{cases} \psi'_{LH2}(x,y)+8 & \text{if } h'_{\psi 1}(j)=\ell+8,\ \varphi(n)=0, \\ \psi'_{LH2}(x,y)+4 & \text{if } h'_{\psi 1}(j)=\ell+4,\ \varphi(n)=1, \\ \psi'_{LH2}(x,y)-8 & \text{if } h'_{\psi 1}(j)=-(\ell+8),\ \varphi(n)=0, \\ \psi'_{LH2}(x,y)-4 & \text{if } h'_{\psi 1}(j)=-(\ell+4),\ \varphi(n)=1, \end{cases} \tag{48}$$

for $1\leq \ell\leq T$.

Step 9: Restore the original histogram. The original histogram $h'''_{\psi 1}$ can be calculated according to the following rules:

$$h'''_{\psi 1}(j)=\begin{cases} h''_{\psi 1}(j)-8 & \text{if } h''_{\psi 1}(j)\geq T+1, \\ h''_{\psi 1}(j)+8 & \text{if } h''_{\psi 1}(j)\leq -(T+1). \end{cases} \tag{41}$$

for $\ell\leq L$.

The restoration of the histogram as described above results in changes in the coefficients. This can be described by the following formula:

$$D'''_{\psi 1}(x,y)=\psi_{HL2}(x,y)-\psi'''_{LH2}(x,y). \tag{42}$$

Here, $\psi'''_{LH2}(x,y)$ can be expressed as:

$$\psi'''_{LH2}(x,y)=\begin{cases} \psi''_{LH2}(x,y)+8, & \text{if } h''_{\psi 1}(j)\geq T+1, \\ \psi''_{LH2}(x,y)-8, & \text{if } h''_{\psi 1}(j)\leq -(T+1). \end{cases} \tag{43}$$

Step 10: After recovering the *LH2* sub-band, steps 1 to 9 are repeated to recover the rest of *HL2* and *HH2* sub-bands as $\psi'''_{HL2}(x,y)$ and $\psi'''_{HH2}(x,y)$.

Obtaining the *LL* Sub-band:

Step 11: Recover the original Sub-band *LL* through the inverse operation of the HDWT algorithm with $\psi_{LL2}(x,y)$, $\psi'''_{LH2}(x,y)$, $\psi'''_{HL2}(x,y)$, and $\psi'''_{HH2}(x,y)$. This procedure can be formulated as follows:

$$LL=IDWT(\psi_{LL2},\psi'''_{LH2},\psi'''_{HL2},\psi'''_{HH2}). \tag{44}$$

Recovering *LH, HL* and *HH* Sub-bands:

Step 12: After recovering the *LL* sub-band, steps 1 to 9 are also repeated to recover the *LH*, *HL* and *HH* sub-bands.

Obtaining Original Image:

Step 13: Recover the original image through the inverse operation of the HDWT algorithm with the *LH*, *HL* and *HH* sub-bands. This procedure can be formulated as follows:

$$\eta=IDWT(LL,LH,HL,HH). \tag{45}$$

These steps finish the extraction and recovering process.

4. Experimental results and comparison

In this section, a series of experiments are performed to evaluate of our scheme. For these experiments, we used many different types of images, including some commonly used ones and two medical images (Fig. 5). The message bits to be embedded in our experiments are randomly generated by a pseudo-random binary generator. The threshold ranges from 0 to 100.

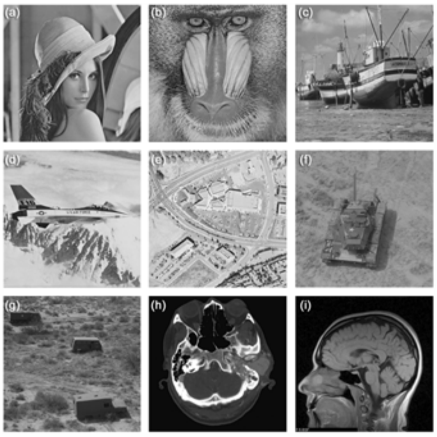


Fig. 5. 8-bit 512×512 images(a)Lena, (b)Baboon, (c)Boat, (d)Airplane, (e)Aerial, (f)Tank, (g)Trucks, (h)Medical image1, (i) Medical image2 .

Capacity versus Threshold

The relationship between the capacity (in bpp) and the threshold is presented in Fig. 6. The embedding rate almost reaches 0.8 bpp at threshold 100 for most the test images. As expected the capacity is nearly proportional to the threshold at the beginning and saturates when the threshold is sufficiently high.

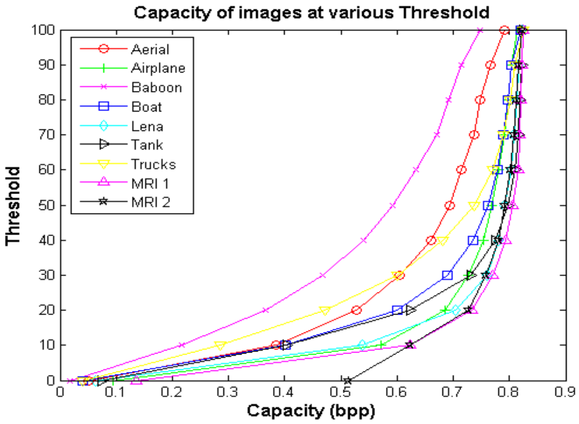


Fig. 6. Embedding capacity at various thresholds.

Visual Quality versus Threshold

Fig. 7 depicts the visual quality in PSNR of the marked image versus threshold varying from 0 to 100 for test images on the premise that maximal bits are embedded. The experimental result indicates that the PSNR rises as the threshold increases and this is not the case for the previous studies. The marked image can achieve 43 dB at the threshold 0, and above 46 dB at the threshold of 100 for most test images. It's noteworthy that larger values of threshold contribute less variation to the histogram of the wavelet transform. Given that the middle/high-frequency sub-bands incorporate less energy, the test images with larger variance between middle and high-wavelet coefficients such as "MRI 1" and "MRI 2" can achieve higher visual quality than "Baboon" at the threshold 0.

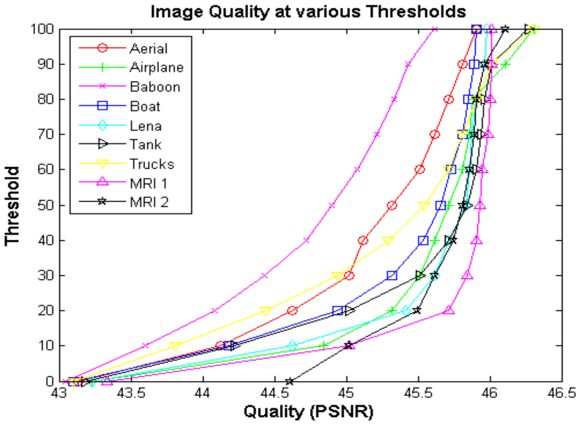
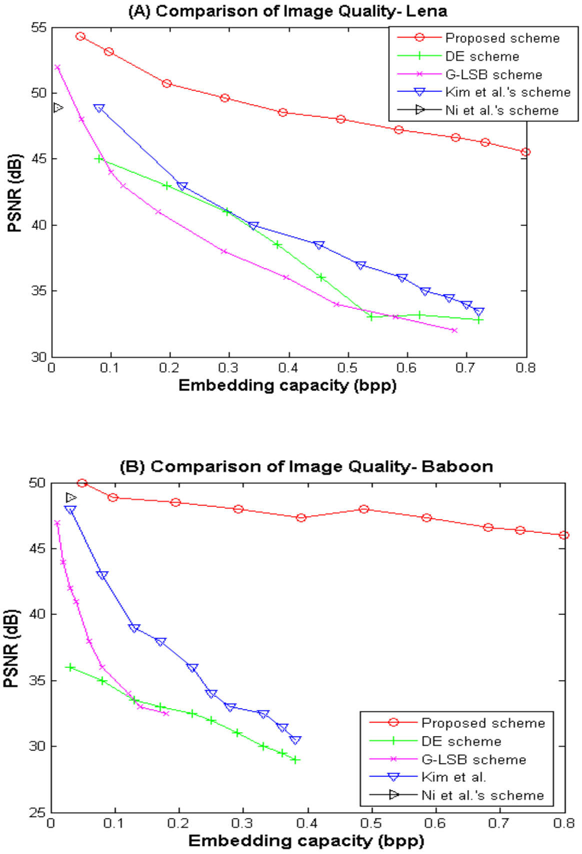


Fig. 7. PSNR at various thresholds.

Comparison of visual quality with other schemes

To address the great contribution of our scheme, analyses in terms of actual embedding capacity and visual quality were performed. The proposed scheme was compared with the DE scheme [14], G-LSB scheme [5], Kim et al.'s scheme [9], and Ni et al.'s scheme [6] for the "Lena", "Baboon", "Boat", and "Airplane" images as shown in Fig. 8. The embedding capacity is the amount of embedded bits with overhead subtracted. It is observed that our proposed technique has achieved the highest PSNR at the same bpp.



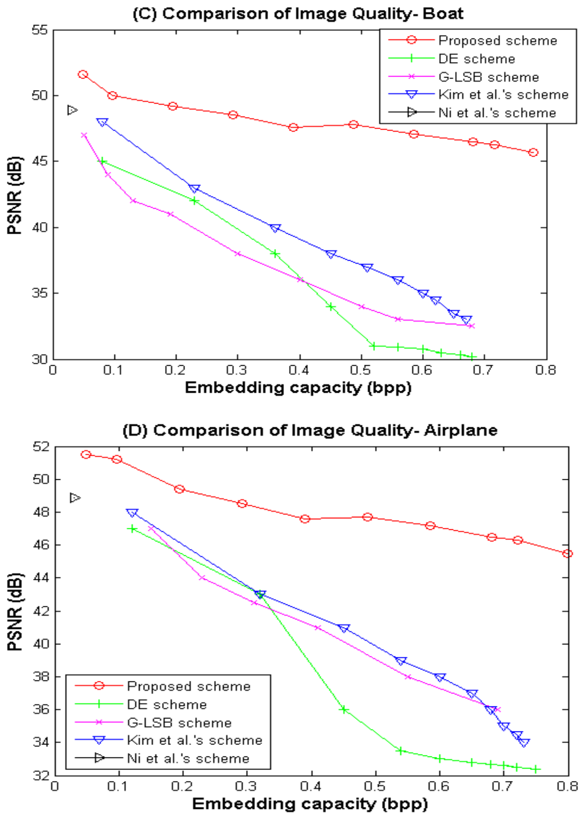


Fig. 8. Comparison of embedding capacity in bpp versus distortion with existing reversible schemes: DE scheme, G-LSB scheme, Kim et al.'s scheme, and Ni et al's scheme: (A) Baboon; (B) Lena; (C) Boat; (D) Airplane.

5. Conclusion

In this paper, a reversible data hiding scheme exploiting the large variance of wavelet coefficients and clever histogram shifting rules is presented. The proposed scheme, compared with previous ones, can obtain better visual quality of the marked image given the same payload. The main reason is that the visual quality of our scheme does not decay with increasing threshold as in the other schemes. In addition, our scheme provides the greatest embedding capacity and is even better than the one published recently [19]. It may be of interest for future research that the threshold predictions, multi-round schemes, and fast algorithms will be explored to meet real-time application requirements.

References

[1] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding", IBM Systems Journal, vol.35, no.3, pp.313–336, 1996.

[2] M. Awrangjeb, "An overview of reversible data hiding", Proc. Sixth International Conf. on Computer and Information Technology, Jahangirnagar University, Bangladesh, pp. 75–79, December 2003.

[3] J. Fridrich, M. Goljan, and R. Du, "Distortion-free data embedding", Proc. 4th Information Hiding Workshop, New York, vol.2137, pp.27–41, Lecture Notes in Computer Science, 2001.

[4] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding—new paradigm in digital watermarking", EURASIP J. Appl. Signal Process. vol.2, pp.185–196, 2002.

[5] G.. Xuan, Y.Q. Shi, J. Chen, J. Zhu, and Z. Ni, W. Su, "Lossless data hiding based on integer wavelet transform", IEEE International Workshop on Multimedia Signal Processing, St. Thomas, Virgin Islands, USA, December 9-11, 2002.

[6] J. Tian, "Reversible data embedding using a difference expansion", IEEE Trans. on Circuits and Systems for Video Technology vol.13, no.8, pp.890–896, 2003.

[7] A.M. Alattar, "Reversible watermark using difference expansion of triplets", Proc. IEEE International Conference on Image Processing, vol.1, Barcelona, Spain, pp. 501–504, September 2003.

[8] A.M. Alattar, "Reversible watermark using difference expansion of quads", Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Montreal, Canada, vol.3, pp.377–380, May 2004.

[9] L. Kamstra and H.J.A.M. Heijmans, "Wavelet techniques for reversible data embedding into images", Centrum voorWiskunde en Informatica Rep. August 2004.

[10] L. Kamstra and H.J.A.M. Heijmans, "Reversible data embedding into images using wavelet techniques and sorting", IEEE Trans. on Image Processing vol.14, no.12, pp.2082–2090, December 2005.

[11] D.M. Thodi, J.J. Rodriguez, "Expansion embedding techniques for reversible watermarking", IEEE Trans. on Image Processing vol.16, no.3, pp.721–730, 2007.

[12] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication", Proc. of the SPIE, Security and Watermarking of Multimedia Contents, vol.4314, San Jose, CA, pp.197–208, January 2001.

[13] M.U. Celik, G. Sharma, and A.M. Tekalp, "Reversible data hiding", Proc. IEEE International Conf. on Image Processing, Rochester, NY, pp.157–160, 2002.

[14] M.U. Celik, G. Sharma, A.M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding", IEEE Trans. on Image Proc. vol.14, no.2, pp.253–266, February 2005.

[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, March 2006.

[16] J. Hwang, J.W. Kim, and J.U. Choi, "A reversible watermarking based on histogram shifting", International Workshop on Digital Watermarking, Lecture Notes in Computer Science, Springer-Verlag, Jeju Island, Korea, vol.4283, pp.348–361, 2006.

[17] W.-C. Kuo, D.-J. Jiang, and Y.-C. Huang, "Reversible data hiding based on histogram", International Conf. on Intelligent Computing, Lecture Notes in Artificial Intelligence, Springer-Verlag, Qing Dao, China, vol.4682, pp.1152–1161, 2007.

[18] K.-S. Kim, M.-J. Lee, H.-Y. Lee, and H.-K. Lee, "Reversible data hiding exploiting spatial correlation between sub-sampled images", Pattern Recognition, vol.42, pp.3083-3096, 2009.

[19] X.-R. Luo, C.-H. Jerry Lin, and T.-L. Yin, "Reversible data hiding based on two-level HDWT coefficient histograms", Advanced Computing: An International Journal, vol.2, no.1, pp.1-16, January 2011.

Received: December 2010. Accepted: March 2011.