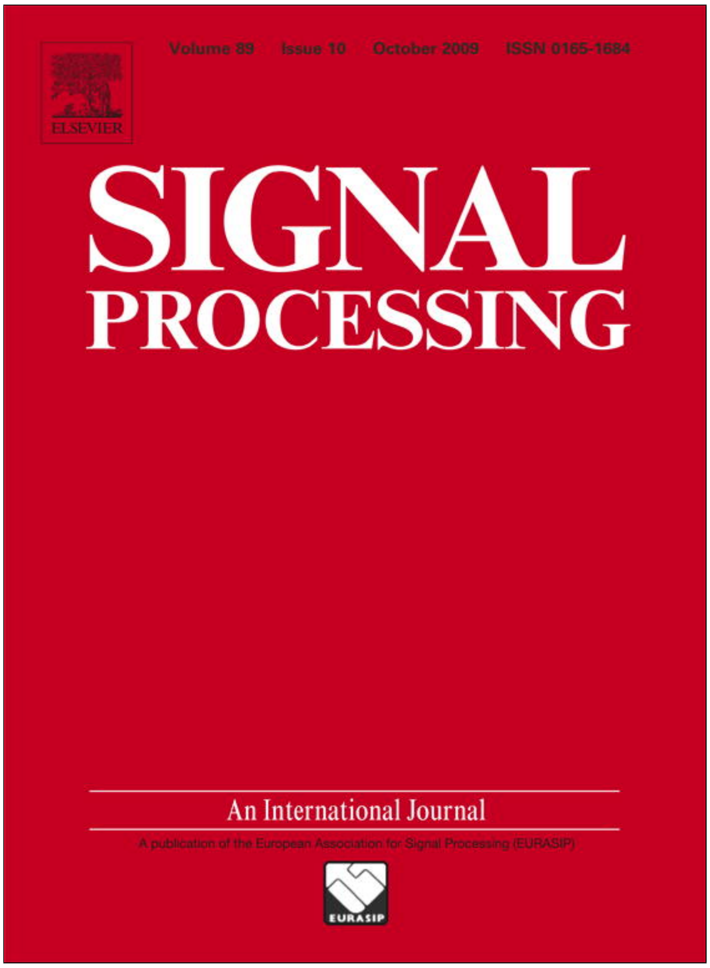
Volume 89 Issue 10 October 2009 ISSN 0165-1684
ELSEVIER
SIGNAL
PROCESSING
An International Journal
A publication of the European Association for Signal Processing (EURASIP)
EURASIP

Signal Processing 89 (2009) 2053–2065

Contents lists available at ScienceDirect

Signal Processing

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

# Reversibility improved lossless data hiding

Xinbo Gao [a,*], Lingling An [a], Xuelong Li [b], Dacheng Tao [c]

[a] School of Electronic Engineering, Xidian University, Xi'an 710071, PR China
[b] School of Computer Science and Information Systems, Birkbeck College, University of London, London WC1E 7HX, UK
[c] School of Computer Engineering, Nanyang Technological University, 50 Nanyang Avenue, Blk N4, 639798, Singapore

ARTICLE INFO

*Article history:*
Received 28 October 2008
Received in revised form
5 April 2009
Accepted 6 April 2009
Available online 23 April 2009

*Keywords:*
Lossless data hiding
Reversibility
Block skipping
Histogram

ABSTRACT

Recently, lossless data hiding has attracted increasing interests. As a reversible watermark scheme, the host media and hidden data should be recovered without distortion. A latest lossless data hiding technique based on image blocking and block classification has achieved good performance for image authentication. However, this method cannot always fully restore all the blocks of host images and watermarks. For this purpose, we propose an improved algorithm, which is characterized by two aspects. *First*, a block skipping scheme (BSS) is developed for the host blocks selection to embed watermark; *secondly*, the embedding level is modified by a novel parameter model to guarantee that the host blocks can be recovered without distortion as well as the embedded data. Extensive experiments conducted on standard grayscale images, medical images, and color images have demonstrated the effectiveness of the improved lossless data hiding scheme.



## 1. Introduction

In recent years, digital multimedia (audio, video and image) have gained widespread applications worldwide, which can be accessed and distributed on the Internet and via CD-ROM with convenience. Meanwhile, due to powerful computer software, the content of the digital multimedia is easy to maliciously modify. Thus "perfect" forgeries are made such that what you observe cannot be trusted [1]. The unrestricted duplication and easy modification have brought to forefront security concerns with the use of digital products. For example, in military communication systems, it needs to identify where a received secret message is from and whether it is maliciously modified. Therefore, copy protection, copyright protection, and content authentication have been the three most important issues in the digital world.

To solve the above problems, data hiding has been proposed and popularly employed for unobtrusive communication, feature tagging, tamper proofing, broadcast monitoring, data augmentation and other purposes [2,21,22]. By embedding secret information, such as digital signature, logo image or identification (ID) number, into digital media, data hiding protects them from unauthorized use. However, traditional data hiding techniques often bring about permanent distortion of the host images, which is not acceptable in some applications such as law enforcement, remote sensing, medical diagnosis, etc. Therefore, lossless data hiding, also called reversible watermarking, has become a hot topic in recent years.

To the best of our knowledge, many lossless data hiding methods have been proposed since the first reversible watermarking technique was designed by Honsinger in 2001 [4]. According to the survey presented by Feng et al. [3], these methods can be classified into three categories: (1) data compression based method; (2) difference expansion based method; and (3) histogram based method. In the first type of methods, Fridrich et al. [5] compressed one of the least significant bit (LSB) planes of the quantized discrete cosine transform (DCT) coefficients to embed image hash. Celik et al. [6] presented a

* Corresponding author. Tel.: +86 29 88201838; fax: +86 29 88201620.
*E-mail address:* xbgao@mail.xidian.edu.cn (X. Gao).

0165-1684/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2009.04.015

lossless generalized-LSB data embedding method which is based on context-based adaptive lossless image coding (CALIC) designed by Wu [7]. Tian [8] utilized difference expansion, a popular technique of the second type, to embed information, which is improved by others for better image quality and higher capacity [9–12]. However, the first and second types of schemes lack robustness, which has limited their potential applications in practice.

In the third type of lossless data hiding methods, De Vleeschouwer et al. [13] presented the first robust reversible watermarking technique, which employed a circular interpretation of bijective transformations together with modulo-256 addition to allow secret data to be appended. However, the algorithm has suffered from severe salt-and-pepper noise for medical images and daily-life images. For this reason, Ni et al. proposed a method based on a robust statistical quantity to eliminate the salt-and-pepper noise [14]. Zou et al. presented another method by using integer wavelet transform (IWT) [18]. All these algorithms utilized the distribution characteristics of the histogram to achieve losslessness with higher invisibility, larger capacity and stronger robustness. Because of the robustness and low computational complexity, the third lossless data hiding schemes have been attracting increasing interests [19,20,23,24]. We focus our interests on this kind of method and extensively investigate the latest method proposed by Ni et al. [14]. This method used a robust statistical quantity and error correction code (ECC) to hide data in host image and achieve losslessness. It has the advantages of no salt-and-pepper noise in the watermarked image, adjustable capacity and a better peak signal-to-noise ratio (PSNR) over the previous work. However, both theoretical analysis and experimental test show that the method is irreversible for almost all of the test images except *Lena*. That is, both the host image and watermark cannot be recovered without distortion. Therefore, we proposed an improved lossless data hiding method, which utilizes a block skipping scheme (BSS) and a novel parameter model to avoid the distortion caused by Ni's method, and thus losslessly recovers the host image as well as watermark.

In the proposed scheme, the host image is first divided into non-overlapped blocks. Each block is classified into four categories according to its histogram distribution characteristics. The BSS is then used to determine whether the block is suitable for embedding watermark. If it is suitable, a watermark bit is embedded into the block; otherwise, it will be skipped. In this process, a novel parameter model and an improved embedding strategy are introduced to guarantee that the host blocks can be recovered without distortion as well as the embedded data. Extensive experimental results show that the proposed scheme improves the reversibility of Ni's method and achieves full lossless recovery for both the host image and watermark.

The rest of this paper is organized as follows. Ni's method [14] and its weakness are reviewed briefly in Section 2. Then, the proposed method is presented in Section 3. In Section 4, theoretical analysis of the improved parts is given, which is followed by experimental results and analysis in Section 5. Finally, conclusions are drawn and the future work is presented in Section 6.

## 2. Ni's method and its weakness

In order to eliminate the annoying salt-and-pepper noise in [13], Ni et al. proposed a lossless image data hiding method based on a robust statistical quantity [14], which uses the high correlation and spatial redundancy between the pixel grayscale values in a local block. First, the host image is divided into non-overlapped blocks. The pixels of each block are split into two sets $A$ and $B$ in the horizontal direction. Let $a_i$ and $b_i$ denote the pixels of sets

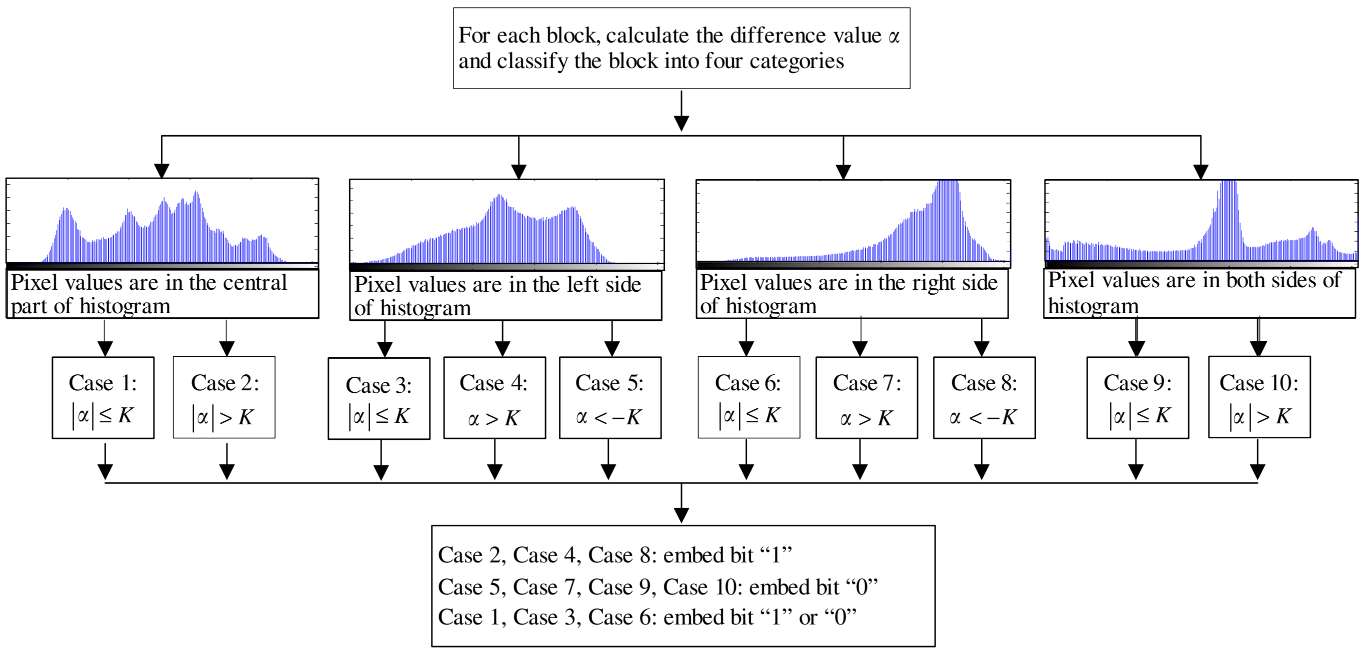


**Fig. 1.** Bit embedding algorithm of Ni's method. $\alpha$ stands for the difference value and $K$ for the threshold.

$A$ and $B$, respectively, which are called a pixel pair. Secondly, a statistical quantity named the difference value $\alpha$ is calculated using the following formula

$$\alpha = \frac{1}{n}\sum_{i=1}^{n}(a_i - b_i), \tag{1}$$

where $n$ is the number of pixel pairs of each block.

Then a watermark bit is embedded into each block according to the bit embedding algorithm shown in Fig. 1. In order to overcome the overflow/underflow problem, different categories use different embedding strategies. That is, bit "1" is always embedded for *Case* 2, *Case* 4 and *Case* 8, and "0" always for *Case* 5, *Case* 7, *Case* 9 and *Case* 10 whenever the to-be-embedded bit is "0" or "1". For other cases, "0" or "1" may be embedded depending on the to-be-embedded bit.

Based on the above procedure, all watermark bits can be embedded into the host image, and thus the watermarked image is generated.

In the watermark extraction process, when the difference value $\alpha$ in the watermarked image is greater than the threshold $K$ and the block does not fall into *Case* 5, *Case* 7 or *Case* 10, bit "1" is extracted; otherwise, bit "0" is extracted. Fig. 2 illustrates the bit extraction process.

Because the embedded bits are fixed in the cases except *Case* 1, *Case* 3 and *Case* 6, they may be different from the to-be-embedded bits. Consequently, the ECC and permutation technique are employed to correct error bits caused by the embedding procedure. In this way, both the watermark and host image are restored.

Ni's method has the advantages of no salt-and-pepper noise in the watermarked image, adjustable capacity and a better PSNR over the previous one. However, our extensive investigation of the embedding and extraction algorithms has revealed two problems facing this method.

(P1) In the extraction process, the host image and watermark cannot be correctly recovered for almost all of the test images except *Lena*, partially because the conditions of block classification are not properly set up. In such a situation, the categories of those blocks, in which pixel values are close to the boundary of two categories, will be changed into *Case* 5, *Case* 7 or *Case* 10 when bit "1" is embedded into them. Thus, the extraction strategies do not match with the embedding ones and errors will result. Two examples are shown in Fig. 3 when the algorithm in [14] is applied to a standard grayscale image, *Boat* ($512 \times 512$), and a color image, *Woman* ($1536 \times 1920$). The PSNRs of the recovered images with respect to the host ones are 43.99 and 43.82 dB, respectively. This means the host images cannot be losslessly recovered. And the image blocks with loss, in which the pixel grayscale values are different from the host ones, are marked by the green boxes. Furthermore, as can be seen from Table 1, 23 "0"s are extracted for the image *Boat*, which means 23 error bits occur in the extraction process.

(P2) The embedding level $\beta$ is defined as twice the threshold $K$ in Ni's method, which is obtained experimentally without theoretical proof. More importantly, it leads to the failure in recovering the host image and watermark when the difference value $\alpha$ equals minus $K$ for *Case* 3 or $K$ for *Case* 6 in the host image, although this case is seldom encountered. Fig. 4 illustrates an example in which the image blocks are restored incorrectly in the recovered image are marked by the green boxes in the same way as Fig. 3.

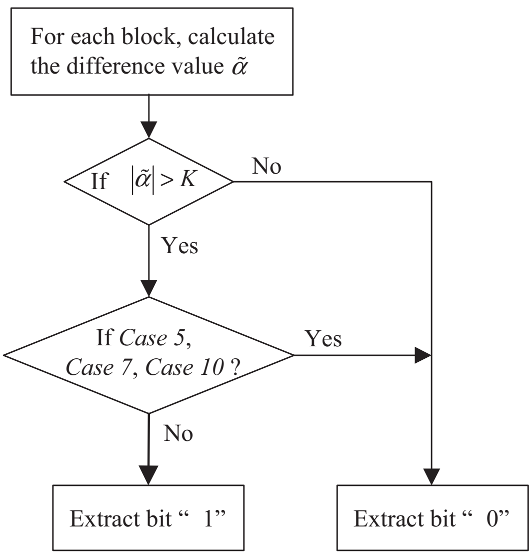


**Fig. 2.** Bit extraction algorithm of Ni's method. $\tilde{\alpha}$ stands for the difference value in the watermarked image. $K$ is the threshold.


**Fig. 3.** (a) Host images, (b) recovered images using Ni's method for *Boat* and *Woman*. The embedding level is 10, and the capacities are 528 bits and 1398 bits, respectively.

Based on the above analysis, an effective method is therefore required to avoid the distortion and thus losslessly recover the host image and watermark.

## 3. The improved lossless data hiding method

For the two problems mentioned above, (P1) can be solved by BSS. That is, it needs to check whether each image block is unstable (which means that this block will not be able to recover without distortion after emmbedding a bit watermark.) or not after embedding a watermark bit. If the block is unstable, which means its category will be changed into *Case* 5, *Case* 7 or *Case* 10, it will be skipped. Otherwise, a watermark bit is embedded into the block. In this way, we can automatically discard all image blocks unsuitable for embedding the watermark, and thus avoid irreversibility caused by the block classification scheme. As for (P2), a novel parameter model together with an improved embedding strategy is introduced to deal with it. In this way the distortion brought about by the data hiding method and parameter definition is avoided. These two schemes form the main contributions of the proposed lossless data hiding method. The new method consists of the watermark embedding procedure and the watermark extraction and image recovery procedure, which are described as follows.

### 3.1. Watermark embedding

The process of watermark embedding is shown in Fig. 5, and the detailed algorithm is presented as follows.

*Step* 1: The host image is first divided into non-overlapped blocks. For each block, the difference value $\alpha$ is calculated with Eq. (1) described in Section 2. And then the block is classified into four categories according to the conditions shown in Fig. 6. It should be noted that this step is the same as the one in Ni's method.

*Step* 2: Whether each block is suitable for embedding watermark is checked using the conditions shown in Table 2. The proof of these conditions is given in Section 4.1. If the conditions are met, this block will be regarded as an unstable block and skipped. Meanwhile, its location in the host image will be recorded. Otherwise, the block is thought to be stable and suitable for embedding watermarks in the next steps. This scheme above is called BSS, by which the distortion caused by the block classification conditions can be avoided.

*Step* 3: The relationship between the threshold $K$ and embedding level $\beta$ is initialized with a novel parameter model given by

$$\beta = tK + \varepsilon, \tag{2}$$

where $t$ is an integer number greater than 1, and $\varepsilon$ is an infinitely small quantity. In this case, the hidden watermarks can be extracted correctly for all cases even if the difference value $\alpha$ equals minus $K$ for *Case* 3 or $K$ for *Case* 6 in the host image. That is, the second problem of Ni's method discussed in Section 2 can be solved successfully. The proof of this part is presented in Section 4.2.

*Step* 4: The stable block is further divided into different cases according to the relationship between $\alpha$ and $K$, and then a watermark bit is embedded into it based on the bit embedding strategy shown in Fig. 7. In order to recover the host image and watermark when the embedding level $\beta$ is twice the threshold $K$ under the proposed parameter model, the conditions of *Case* 3, *Case* 5, *Case* 6 and *Case* 7 are modified.

In the embedding process, if "0" is embedded, the pixel grayscale values of the image block remain intact; otherwise, the difference value $\alpha$ is shifted towards the left-hand side or right-hand side depending on the cases illustrated in Fig. 8. For example, when $\alpha > 0$ for *Case* 1 and *Case* 2, $\alpha$ is shifted towards the right-hand side to embed bit "1". Shifting $\alpha$ towards the left-hand side means subtracting a fixed embedding level, $\beta$, from each pixel grayscale value in set $A$; shifting $\alpha$ towards the right-hand side means adding $\beta$ to all pixels in set $A$.

**Table 1**
Original and extracted watermarks using Ni's method for *Boat* and *Woman*. The original watermarks are a series of "1". In the extraction process, "0" is extracted rather than "1" when an error occurs.

| Images | Original watermarks | Extracted watermarks |
|---|---|---|
| *Boat* (512 × 512) | $\underbrace{111\cdots111}_{528\ \text{“1”}}$ | $\underbrace{110\cdots101}_{505\ \text{“1”}\ and\ 23\ \text{“0”}}$ |
| *Woman* (1536 × 1920) | $\underbrace{111\cdots111}_{1398\ \text{“1”}}$ | $\underbrace{111\cdots111}_{1398\ \text{“1”}}$ |


**Fig. 4.** (a) Host image, (b) recovered image using Ni's method for *Cpic*2. The threshold is 2 and capacity is 1398 bits.

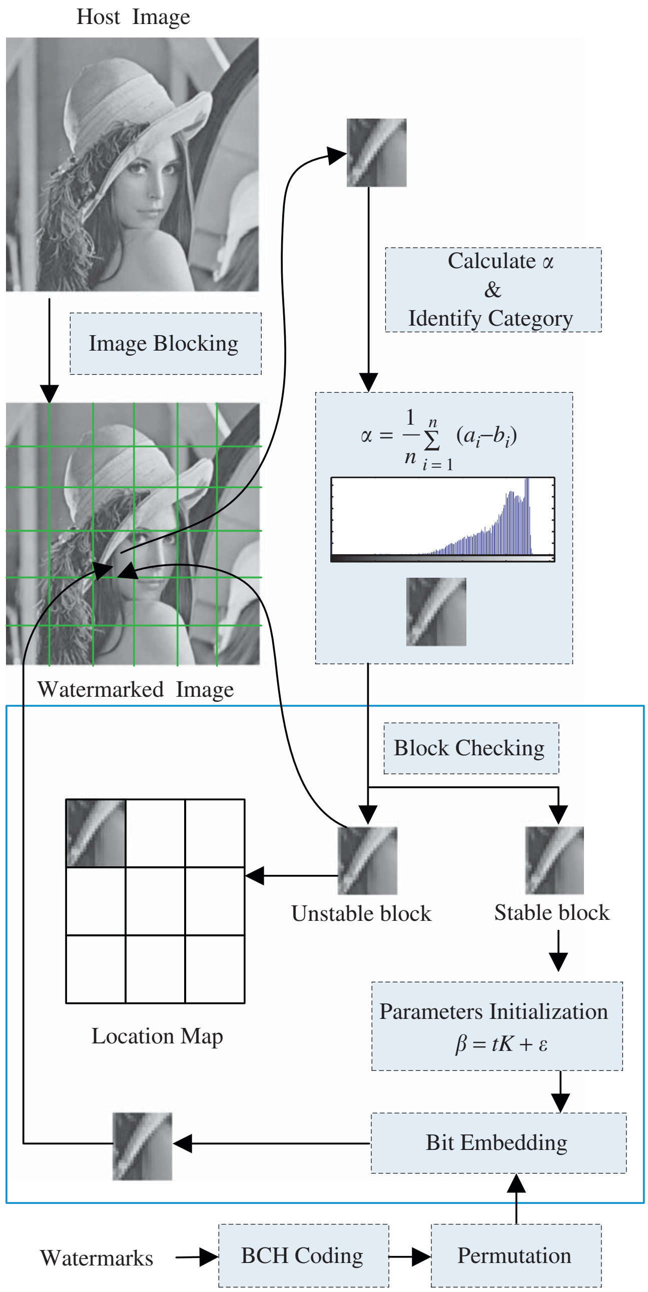


**Fig. 5.** Framework of the proposed watermark embedding scheme. The part enclosed by the blue solid line stands for our improved parts. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Based on the steps mentioned above, the watermarked image can be obtained when all of the watermark bits are embedded into the image.

### 3.2. *Watermark extraction and image recovery*

The procedure for watermark extraction and image recovery is illustrated in Fig. 9. Detailed steps are as follows:

*Step* 1: This step is identical to *Step* 1 in the watermark embedding procedure.

*Step* 2: Each block is checked using the provided location map in which all unstable blocks are recorded. If the block is unstable, it will be skipped. Otherwise, go to the next step.

*Step* 3: The hidden watermark bit is extracted from the block. The extraction method is the same as the one in [14] shown in Fig. 2. If bit "1" is extracted, the difference value $\alpha$ is shifted towards the direction opposite to that of the embedding process to recover the host image. If bit "0" is extracted, $\alpha$ remains constant.

When all of the watermark bits are extracted correctly, the host image can be recovered without distortion.

## 4. Theoretical analysis of the improved parts

As stated above, BSS and a novel parameter model are introduced in the proposed method to deal with the two problems in [14]. The BSS is employed to select unstable blocks and discard them, while the novel parameter model to define the relationship between the embedding level and the threshold. However, the question arises: how these improvements can successfully avoid the distortion and thus achieve a lossless recovery of the host image and watermark. In this section, we will provide a theoretical analysis of the improved parts to validate their effectiveness.

### 4.1. *Proof of block skipping scheme*

As mentioned above, when the category of the block is changed into *Case* 5, *Case* 7 or *Case* 10 from others after embedding bit "1", an error occurs in the watermark extraction. Thus, in the proposed method this block will be regarded as unstable blocks and skipped. Also, the block checking conditions are illustrated in Table 2. Then, in Theorem 1 which follows, it will be shown that these conditions are sufficient to select unstable blocks exactly.

**Theorem 1.** *Suppose I is an image block, max and min are its maximum and minimum pixel values, respectively. Let A represent a pixel set in which each pixel is selected from I in the horizontal direction.* $\overline{max}$ *and* $\overline{min}$ *denote the maximum and minimum pixel values in set A, and* $\beta$ *is the embedding level. If the pixel values of the block meet the conditions shown in Table 2 in Section* 3.1, *then its category will be changed in the way illustrated in Fig.* 10 *when bit* "1" *is embedded into it.*

**Proof.** Suppose $\alpha > 0$ and the category is *Category* 1 which means the case may be *Case* 1 or *Case* 2, we prove Theorem 1 as follows.

Let $\widetilde{max}$ denote the maximum pixel value in the block when bit "1" is embedded. $\widetilde{d}_r$ means the difference between $\widetilde{max}$ and 255. According to Fig. 8 in Section 3.1, when the case is *Case* 1 or *Case* 2 and $\alpha > 0$, $\alpha$ is shifted towards the right-hand side to embed bit "1". That is, $\beta$ is added to all pixels in set *A*. Thus, in order to determine the category of block *I* after embedding, there are two cases

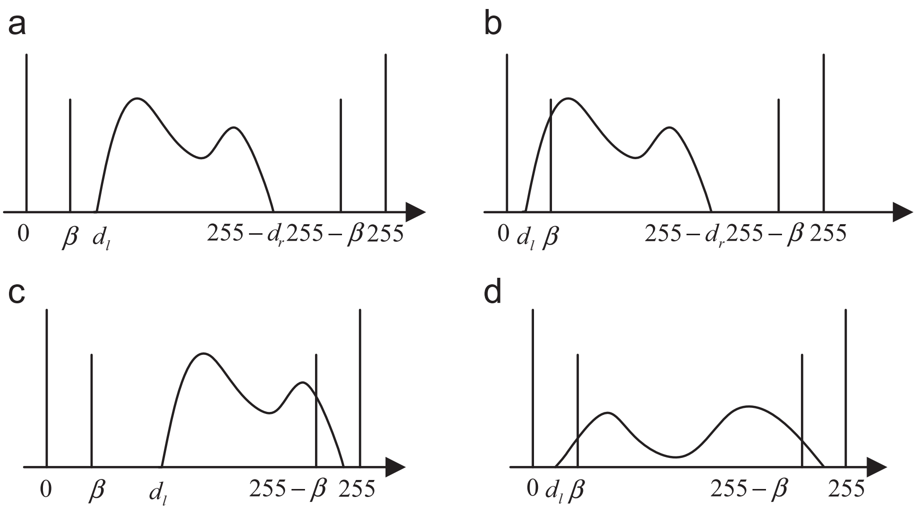


**Fig. 6.** Pixel grayscale values distribution and classification conditions of each category. *dl* stands for the difference between the minimum pixel value and 0, and *dr* stands for the difference between the maximum pixel value and 255. $\beta$ is the embedding level. (a) *Category* 1: $dl \geqslant \beta$ and $dr \geqslant \beta$, (b) *Category* 2: $dl \leqslant \beta$ and $dr \geqslant \beta$, (c) *Category* 3: $dl \geqslant \beta$ and $dr \leqslant \beta$, (d) *Category* 4: $dl \leqslant \beta$ and $dr \leqslant \beta$.

**Table 2**
Block checking conditions for *Category* 1, *Category* 2, and *Category* 3.

| Category | Conditions |
|---|---|
| *Category* 1 | |
| $\alpha > 0$ | $(\exists max \in A \cap max > 255 - 2\beta) \cup$ $(\forall max \notin A \cap \overline{max} + \beta > max \cap \overline{max} > 255 - 2\beta)$ |
| $\alpha \leqslant 0$ | $(\exists min \in A \cap min < 2\beta) \cup$ $(\forall min \notin A \cap \overline{min} - \beta < min \cap \overline{min} < 2\beta)$ |
| *Category* 2 | The same as *Category* 1 when $\alpha > 0$ |
| *Category* 3 | The same as *Category* 1 when $\alpha \leqslant 0$ |

*max* and *min* are the maximum and minimum pixel values in the block, respectively. $\overline{max}$ and $\overline{min}$ are the maximum and minimum pixel values in set *A*, where *A* denotes a pixel set in which each pixel is selected in the horizontal direction. For details about the selection of set *A*, refer to [14]. $\beta$ is the embedding level.

we should consider depending on the relationship between *max* and *A*.

(1) If $max \in A$, $\widetilde{max}$ is given by $\widetilde{max} = max + \beta$. And $max > 255 - 2\beta$, thus $\widetilde{max} > 255 - \beta$, and then $\tilde{d}_r = 255 - \widetilde{max} < \beta$.
(2) If $max \notin A$ and $\overline{max} + \beta > max$ hold, $\widetilde{max}$ can be written as $\widetilde{max} = \overline{max} + \beta$. And $\overline{max} > 255 - 2\beta$, thus $\tilde{d}_r = 255 - \widetilde{max} < \beta$.

As discussed above, $\tilde{d}_r < \beta$ holds for the two cases. From Fig. 6, it can be seen that the category of the given block will be transferred from *Category* 1 to *Category* 3. In addition, since $\beta > K$ deduced from Eq. (2), the following formula is yielded

$$\alpha + \beta > K \tag{3}$$

with $\alpha > 0$. That is to say, the difference value $\alpha$ is greater than *K* after embedding. Therefore, according to Fig. 7, the case of the given block after embedding bit "1" is only transferred to *Case* 7.

What is discussed above is the first condition in Table 2. As for other conditions, a similar procedure can be used to prove them. In a word, using Theorem 1, the unstable blocks, in which the category of the block is changed into *Case* 5, *Case* 7 or *Case* 10 from others after embedding bit "1", are selected exactly. This proves the effectiveness of BSS. □

### 4.2. Proof of parameter model

According to the watermark extraction algorithm shown in Fig. 2, the absolute value of the difference value $\alpha$ should be greater than the threshold *K* after embedding bit "1" for all cases except *Case* 5, *Case* 7 and *Case* 10. In this way, the embedded bit "1" and the host image can be recovered without distortion. However, in [14], when the difference value $\alpha$ equals minus *K* for *Case* 3 or *K* for *Case* 6 in the host image, the above requirement is not met and thus errors occur. Therefore, a novel parameter model and an improved embedding algorithm are proposed to solve this problem. In the following Theorem 2, we will show that the above requirement can be met in the proposed method.

**Theorem 2.** *Suppose* $\beta = tK + \varepsilon$, *where* $t = 2, 3, 4, \ldots, N$, $\varepsilon = 0, 1, 2, \ldots$, *and K is the threshold, the following formula*

$$|\alpha \pm \beta| > K \tag{4}$$

*always holds for different cases, as shown in Table 3.*

**Proof.** *Case* 1 is considered first. Because $0 < \alpha \leqslant K$, $\alpha + \beta$ can be given by $\beta < \alpha + \beta \le K + \beta$. And $\beta = tK + \varepsilon$ holds, thus $tK + \varepsilon < \alpha + \beta \leqslant (t+1)K + \varepsilon$. Substituting $t = 2$ and $\varepsilon = 0$ into it, the minimum of $\alpha + \beta$ can be obtained as $2K < \alpha + \beta \leqslant 3K$. That is, Eq. (4) holds. Likewise, the theorem can be proved by a similar process for other cases. □

Based on Theorem 2 and the embedding algorithm, it can be found that the difference value $\alpha$ after embedding

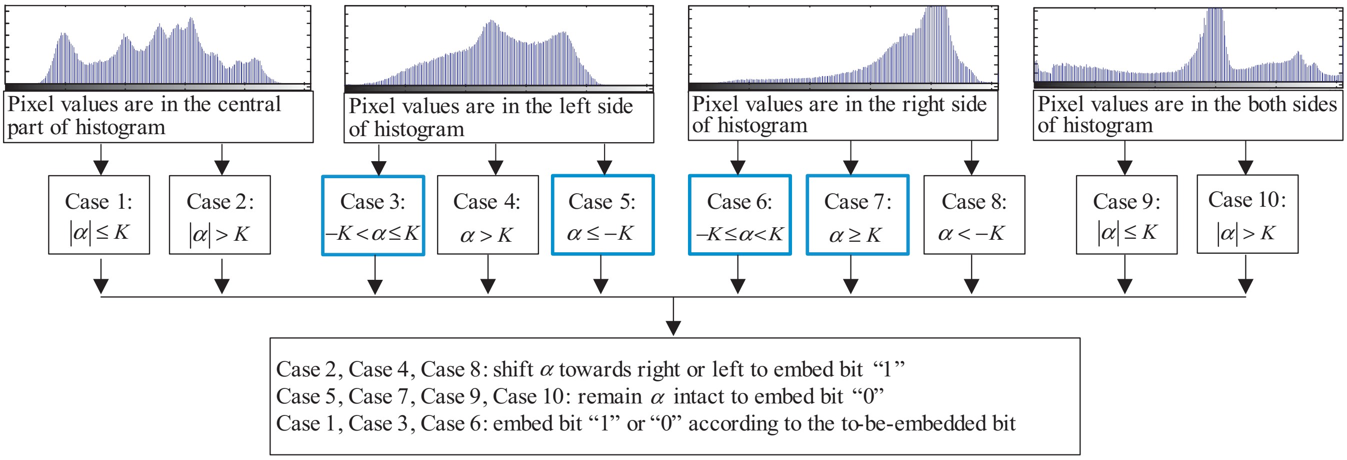


**Fig. 7.** Framework of the improved bit embedding scheme. The improved parts are enclosed by four blue solid lines, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

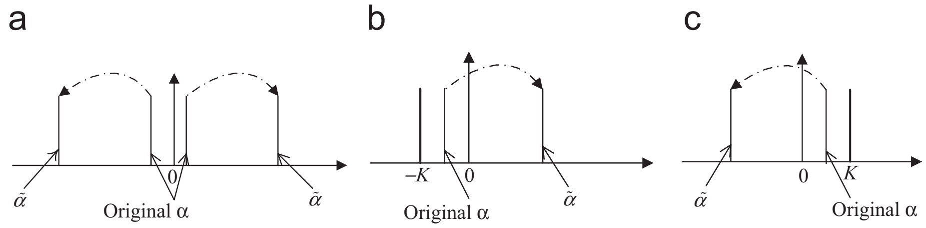


**Fig. 8.** Direction of shifting the difference value $\alpha$ when embedding bit "1" for (a) *Case* 1 or *Case* 2, (b) *Case* 3 or *Case* 4, and (c) *Case* 6 or *Case* 8. $\alpha$ stands for the original difference value and $\tilde{\alpha}$ for the shifted one.

"1" will be greater than the threshold $K$. Consequently, the novel parameter model is proved.

## 5. Experimental results and analysis

In order to validate the effectiveness of the proposed lossless data hiding algorithm, four groups of experiments are conducted in comparison with the method in [14]. These experiments are carried out upon the three standard grayscale images, *Lena*, *Baboon*, and *Boat*, two medical images and three color test images. Note that the images are what Ni et al. adopted. For color images, only the red color plane is applied by the algorithm. The watermark is a series of "1" and the capacity and block size of each image are shown in Table 4, in which the block size is meant the number of pixels along one side of the square block. Also, in order to make a fair comparison, we adopt the same embedding level as that of [14]. That is, $\beta$ is twice the threshold $K$ and $K$ ranges from 1 to 5.

### *5.1. Block skipping scheme test*

In the proposed method, the BSS is designed to find out unstable blocks in the host image. Fig. 11 illustrates the results of block checking for all test images when the threshold $K$ is 5. The selected unstable blocks are marked by the green boxes. In *Lena*, there are no unstable blocks, which is the reason why it can only be recovered without distortion in the eight test images for Ni's method. Then, the question that follows is whether these unstable blocks are the same as the image blocks restored incorrectly in [14]. The experiments in this section are designed to explore this question from two aspects: the number and location of blocks.

#### *5.1.1. Test of number of unstable blocks*

Here, we calculate the number of unstable blocks of method in [14] and our method, respectively. First, we count the case occurrence in the host image and water-marked image, and thus calculate the number of the blocks whose categories have been changed into *Case* 5, *Case* 7, or *Case* 10 after embedding "1", which is the number of the blocks restored incorrectly in [14]. The results of all test images are shown in Tables 5–7, in which errors indicate the numbers of the blocks restored incorrectly. With the increase of errors, the distortion in the recovered image is increased.

Secondly, by calculating the size of the location map generated in the embedding process, we can obtain the number of unstable blocks in our method. Tables 8 and 9 show the comparison of these numbers calculated in two different ways above. As shown in Tables 8 and 9, the same numbers are obtained for all test images, which show that the number of the unstable blocks found out by BSS is same as that of the blocks restored incorrectly in [14].

#### *5.1.2. Test of location of unstable blocks*

To consider whether or not the locations of unstable blocks are same as those restored incorrectly in [14], we

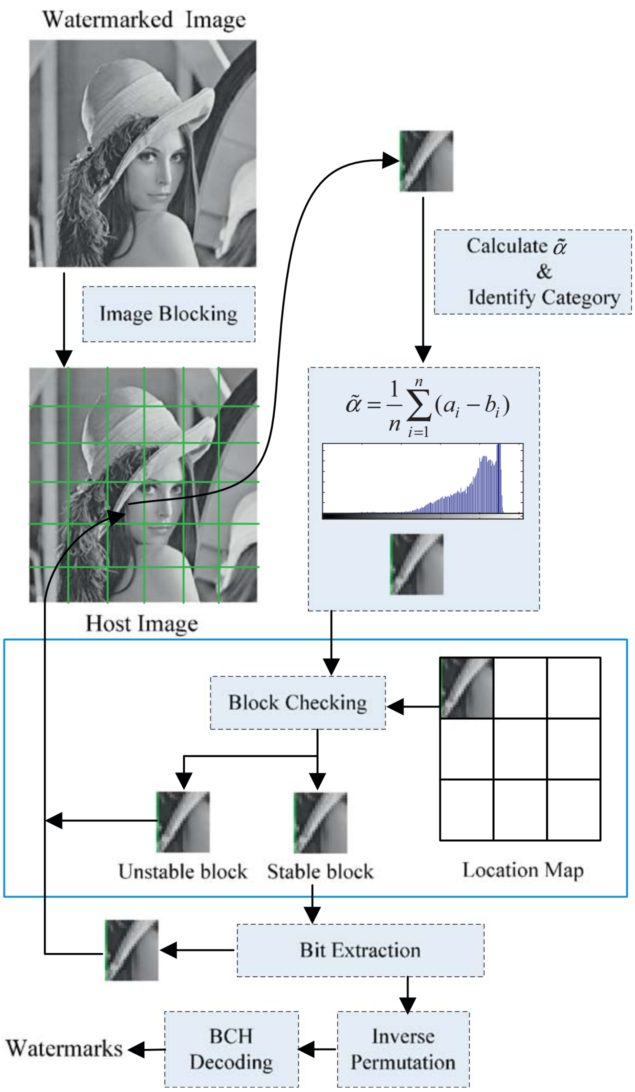


**Fig. 9.** Framework of the proposed watermark extraction and image recovery scheme. The part enclosed by the blue solid line stands for our improved parts. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

first mark the blocks restored incorrectly in [14] by calculating PSNR of the recovered image with respect to the host one. Then, these locations are compared with the location map recorded by the proposed scheme. Fig. 12 shows the result of *Baboon* at different embedding levels, from which it can be seen that the locations are identical. As for other test images, the same results are achieved.

From the above experimental comparison, the unstable blocks selected by the proposed BSS are indeed the same as the image blocks restored incorrectly in [14]. Therefore, the BSS is effective in solving the first problem of [14] discussed in Section 2.

### 5.2. Parameter model test

As analyzed, when the difference value $\alpha$ equals minus $K$ for *Case* 3 or $K$ for *Case* 6 in the host image, errors will occur. Therefore, we define a novel parameter model and modify the embedding algorithm to deal with this problem. In order to validate the effectiveness of this

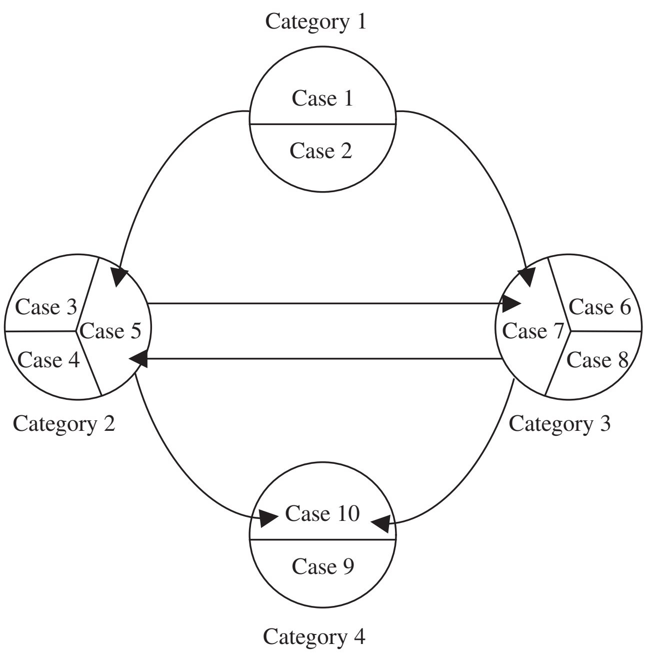


**Fig. 10.** Diagram of category transference when bit "1" is embedded into the block. Note that the arrow represents the direction and terminal of transference. For example, the upper left arrow means the category is changed from *Category* 1 to *Case* 5 of *Category* 2.

model, we count the number of blocks in which $\alpha$ equals minus $K$ for *Case* 3 or $K$ for *Case* 6 in the host image. The experimental results are shown in Tables 10 and 11.

It can be found that there are no blocks which meet the conditions above in our method. This shows that the parameter model is successful in solving the second problem in [14]. Note that the pixel values in a block are often highly correlated, so the difference value $\alpha$ is very close to zero. That means errors often occur when the threshold $K$ is small. With the increase of $K$, errors will disappear. In addition, since such errors are only related to *Case* 3 and *Case* 6, there are no errors when the block is in other cases except *Case* 3 and *Case* 6.

Based on the above improvements, the host image and watermark can be restored without distortion using the proposed method. Tables 12 and 13 show the recovery of the watermark. In most cases, the hidden data can also be correctly retrieved in [14]. This is because ECC and the permutation technique are used to correct errors caused by the embedding algorithm. Moreover, all test images can be completely restored in our method, while only *Lena* can be recovered in [14] because there are unstable blocks in test images except *Lena*.

### 5.3. Effect on invisibility

Typically, there are two factors which influence invisibility: embedding capacity and embedding level. The embedding level represents the amount of grayscale value change for each pixel when either bit "1" or "0" is embedded into the image. Given a fixed embedding level, invisibility is decreased with the increase of embedding capacity. And given a fixed embedding capacity, the higher

**Table 3**
Values of $\alpha \pm \beta$ for different cases.

| Case | $\alpha$ | $\alpha \pm \beta$ | $\alpha \pm \beta$ ($t=2$ and $\varepsilon=0$) |
|---|---|---|---|
| *Case* 1 | $0<\alpha\leqslant K$ | $tK+\varepsilon<\alpha+\beta\leqslant(t+1)K+\varepsilon$ | $2K<\alpha+\beta\leqslant 3K$ |
| | $-K\leqslant\alpha\leqslant 0$ | $-(t+1)K-\varepsilon\leqslant\alpha-\beta\leqslant -tK-\varepsilon$ | $-3K\leqslant\alpha-\beta\leqslant -2K$ |
| *Case* 2 | $\alpha>K$ | $\alpha+\beta>(t+1)K+\varepsilon$ | $\alpha+\beta>3K$ |
| | $\alpha<-K$ | $\alpha-\beta<-(t+1)K-\varepsilon$ | $\alpha+\beta<-3K$ |
| *Case* 3 | $-K<\alpha\leqslant K$ | $(t-1)K+\varepsilon<\alpha+\beta\leqslant(t+1)K+\varepsilon$ | $K<\alpha+\beta\leqslant 3K$ |
| *Case* 4 | $\alpha>K$ | $\alpha+\beta>(t+1)K+\varepsilon$ | $\alpha+\beta>3K$ |
| *Case* 6 | $-K\leqslant\alpha<K$ | $-(t+1)K-\varepsilon\leqslant\alpha-\beta<(1-t)K-\varepsilon$ | $-3K\leqslant\alpha-\beta<-K$ |
| *Case* 8 | $\alpha<-K$ | $\alpha-\beta<-(t+1)K-\varepsilon$ | $\alpha-\beta<-3K$ |

**Table 4**
Embedding capacity and block size of each test image.

| | *Lena* | *Baboon* | *Boat* | *Mpic*1 | *Mpic*2 | *Cpic*1 | *Cpic*2 | *Cpic*3 |
|---|---|---|---|---|---|---|---|---|
| Capacity (bits) | 792 | 560 | 528 | 100 | 100 | 1296 | 763 | 763 |
| Block size | 8 | 12 | 18 | 8 | 8 | 20 | 20 | 20 |

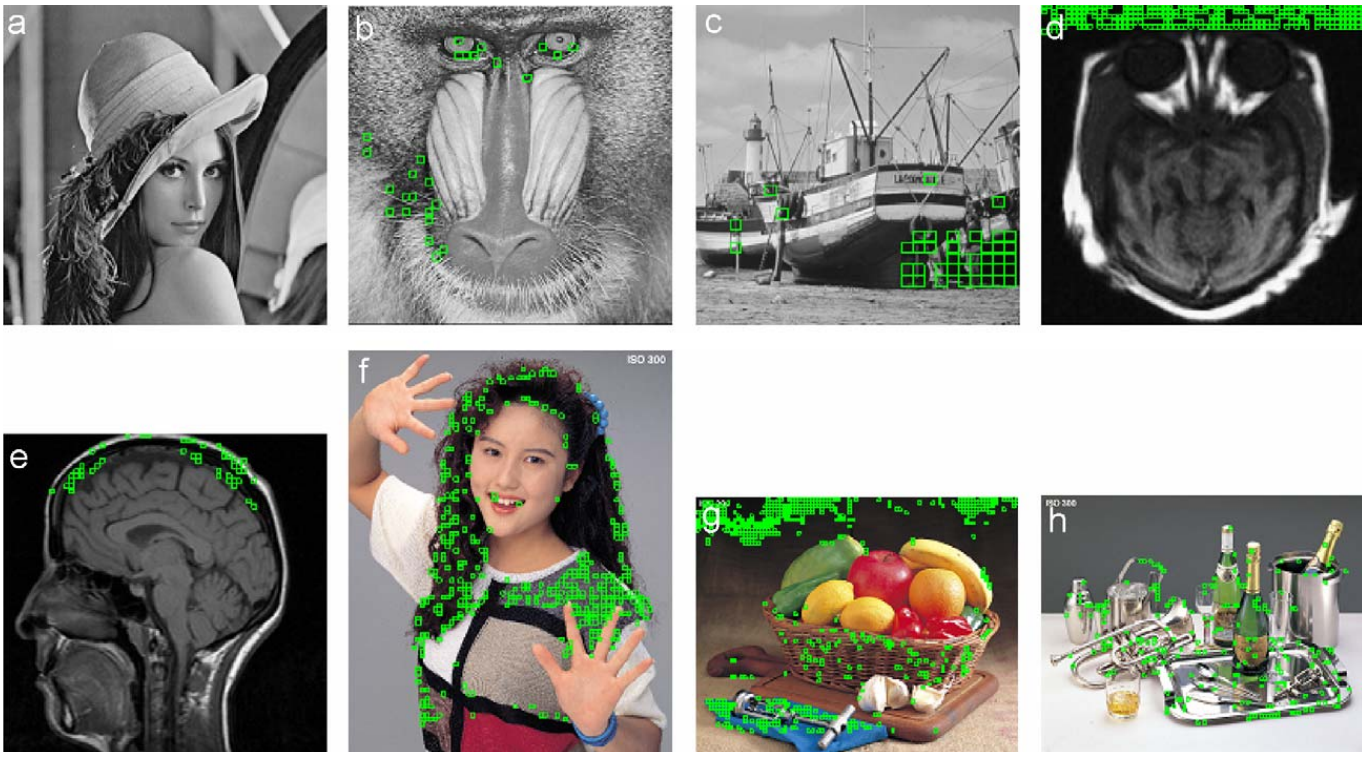


**Fig. 11.** Results of block checking for (a) *Lena*, (b) *Baboon*, (c) *Boat*, (d) *Mpic*1, (e) *Mpic*2, (f) *Cpic*1, (g) *Cpic*2, and (h) *Cpic*3.

the embedding level is, the lower the invisibility is, and vice versa.

In our experiments, in order to measure the effect of our method on invisibility, the PSNRs between the watermarked images and host ones at different embedding levels are calculated, shown in Tables 14 and 15. It can be seen that invisibility is improved at a fixed embedding level because unstable blocks in an image are not used to embed watermarks and thus the capacity is decreased. But it is improved just a little because the number of unstable blocks is small in most cases.

### 5.4. *Effect on embedding capacity*

The experiment in this section is designed to explore the effect of the proposed method on embedding capacity. Fig. 13 illustrates the comparison of capacity with different block sizes when the two algorithms are applied to three standard grayscale images, *Lena*, *Baboon*, and *Boat*. It can be seen that the capacity is decreased just a little for the same reason as that with the improvement of invisibility. In addition, the embedding capacity can be adjusted according to the block size. The larger the block

**Table 5**
Case occurrence and number of error blocks of standard grayscale images in [14].

| Embedding level | Images | *Lena* | | | | *Baboon* | | | | *Boat* | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | *Case* 5 | Case 7 | *Case* 10 | Error | *Case* 5 | *Case* 7 | *Case* 10 | Error | *Case* 5 | *Case* 7 | *Case* 10 | Error |
| 2 | Host | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Watermarked | 0 | 0 | 0 | | 0 | 0 | 0 | | 0 | 0 | 0 | |
| 4 | Host | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| | Watermarked | 0 | 0 | 0 | | 1 | 0 | 0 | | 1 | 0 | 0 | |
| 6 | Host | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 1 |
| | Watermarked | 0 | 0 | 0 | | 3 | 0 | 0 | | 1 | 0 | 0 | |
| 8 | Host | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | 0 | 0 | 0 | 15 |
| | Watermarked | 0 | 0 | 0 | | 15 | 0 | 0 | | 15 | 0 | 0 | |
| 10 | Host | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 26 | 0 | 0 | 0 | 42 |
| | Watermarked | 0 | 0 | 0 | | 26 | 0 | 0 | | 42 | 0 | 0 | |

**Table 6**
Case occurrence and number of error blocks of color images in [14].

| Embedding level | Images | *Cpic*1 | | | | *Cpic*2 | | | | *Cpic*3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | *Case* 5 | *Case* 7 | *Case* 10 | Error | *Case* 5 | *Case* 7 | *Case* 10 | Error | *Case* 5 | *Case* 7 | *Case* 10 | Error |
| 2 | Host | 14 | 1 | 2 | 348 | 6 | 4 | 8 | 33 | 20 | 7 | 70 | 20 |
| | Watermarked | 352 | 3 | 10 | | 32 | 5 | 14 | | 27 | 8 | 82 | |
| 4 | Host | 1 | 1 | 1 | 315 | 0 | 2 | 4 | 87 | 2 | 1 | 18 | 49 |
| | Watermarked | 295 | 3 | 20 | | 69 | 3 | 21 | | 21 | 2 | 47 | |
| 6 | Host | 0 | 0 | 0 | 289 | 0 | 1 | 2 | 176 | 1 | 0 | 7 | 83 |
| | Watermarked | 259 | 5 | 25 | | 149 | 2 | 28 | | 29 | 2 | 60 | |
| 8 | Host | 0 | 0 | 0 | 311 | 0 | 1 | 1 | 283 | 0 | 0 | 1 | 126 |
| | Watermarked | 276 | 6 | 29 | | 243 | 2 | 40 | | 40 | 2 | 85 | |
| 10 | Host | 0 | 0 | 0 | 386 | 0 | 1 | 0 | 563 | 0 | 0 | 1 | 219 |
| | Watermarked | 339 | 9 | 38 | | 512 | 2 | 50 | | 66 | 2 | 152 | |

**Table 7**
Case occurrence and number of error blocks of medical images in [14].

| Embedding level | Images | *Mpic*1 | | | | *Mpic*2 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | *Case* 5 | *Case* 7 | *Case* 10 | Error | *Case* 5 | *Case* 7 | *Case* 10 | Error |
| 2 | Host | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 69 |
| | Watermarked | 0 | 0 | 0 | | 69 | 0 | 0 | |
| 4 | Host | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 50 |
| | Watermarked | 4 | 0 | 0 | | 50 | 0 | 0 | |
| 6 | Host | 0 | 0 | 0 | 278 | 0 | 0 | 0 | 44 |
| | Watermarked | 278 | 0 | 0 | | 44 | 0 | 0 | |
| 8 | Host | 0 | 0 | 0 | 317 | 0 | 0 | 0 | 46 |
| | Watermarked | 317 | 0 | 0 | | 46 | 0 | 0 | |
| 10 | Host | 0 | 0 | 0 | 245 | 0 | 0 | 0 | 47 |
| | Watermarked | 245 | 0 | 0 | | 47 | 0 | 0 | |

is, the smaller the capacity is. This is the same as Ni's method. To sum up, the proposed method keeps the advantage of capacity adjustment, although the capacity is reduced a little.

## 6. Conclusions

Compared with invisibility and embedding capacity, reversibility is a more essential criterion in lossless data hiding. Therefore, we extensively analyze the latest lossless data hiding technique based on the histogram proposed by Ni et al. [14] and make an improvement of its reversibility. It utilizes BSS to exactly select all blocks unsuitable for embedding watermarks and discard them, and thus to effectively avoid the distortion caused by block classification conditions. Meanwhile, a novel parameter model together with an improved embedding category is proposed to eliminate the effect of parameter definition and data hiding strategies on image and watermark recovery. Consequently, the proposed method not only recovers the host image and watermark without distortion, but also has the same merits as the previous one, i.e. (1) no salt-and-pepper noise, (2) capacity adjustment. Experimental results upon standard grayscale images, medical images and color images have demonstrated these advantages. Whereas, a clear disadvantage of our algorithm is that the locations of all unstable blocks need to be provided in the extraction process. Therefore, we plan to make a study of how to achieve lossless recovery without a location map, make a comparison of our method with other lossless data hiding methods in the future, extend the scheme for local feature based watermarking [15–17], and generalize the scheme for tensor data representations [25,26].

**Table 8**
Comparison of the number of unstable blocks.

| Embedding level | *Lena* | | *Baboon* | | *Boat* | | *Mpic*1 | |
|---|---|---|---|---|---|---|---|---|
| | Ours | Ni's [14] | Ours | Ni's [14] | Ours | Ni's [14] | Ours | Ni's [14] |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 1 | 1 | 1 | 1 | 4 | 4 |
| 6 | 0 | 0 | 3 | 3 | 1 | 1 | 278 | 278 |
| 8 | 0 | 0 | 15 | 15 | 15 | 15 | 317 | 317 |
| 10 | 0 | 0 | 26 | 26 | 42 | 42 | 245 | 245 |

**Table 9**
Comparison of the number of unstable blocks.

| Embedding level | *Cpic*1 | | *Cpic*2 | | *Cpic*3 | | *Mpic*2 | |
|---|---|---|---|---|---|---|---|---|
| | Ours | Ni's [14] | Ours | Ni's [14] | Ours | Ni's [14] | Ours | Ni's [14] |
| 2 | 348 | 348 | 33 | 33 | 20 | 20 | 69 | 69 |
| 4 | 315 | 315 | 87 | 87 | 49 | 49 | 50 | 50 |
| 6 | 289 | 289 | 176 | 176 | 83 | 83 | 44 | 44 |
| 8 | 311 | 311 | 283 | 283 | 126 | 126 | 46 | 46 |
| 10 | 386 | 386 | 563 | 563 | 219 | 219 | 47 | 47 |

**Table 10**
Number of blocks when $\alpha$ equals minus $K$ for *Case* 3 or $K$ for *Case* 6 in the host image.

| Embedding level | *Lena* | | *Baboon* | | *Boat* | | *Mpic*1 | |
|---|---|---|---|---|---|---|---|---|
| | Ours | Ni's [14] | Ours | Ni's [14] | Ours | Ni's [14] | Ours | Ni's [14] |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 11**
Number of blocks when $\alpha$ equals minus $K$ for *Case* 3 or $K$ for *Case* 6 in the host image.

| Embedding level | *Cpic*1 | | *Cpic*2 | | *Cpic*3 | | *Mpic*2 | |
|---|---|---|---|---|---|---|---|---|
| | Ours | Ni's [14] | Ours | Ni's [14] | Ours | Ni's [14] | Ours | Ni's [14] |
| 2 | 0 | 247 | 0 | 264 | 0 | 194 | 0 | 0 |
| 4 | 0 | 13 | 0 | 9 | 0 | 25 | 0 | 0 |
| 6 | 0 | 2 | 0 | 2 | 0 | 2 | 0 | 0 |
| 8 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 12**
Comparison of recovery of watermark.

| Embedding level | *Lena* | | *Baboon* | | *Boat* | | Mpic1 | |
|---|---|---|---|---|---|---|---|---|
| | Ours | Ni's [14] | Ours | Ni's [14] | Ours | Ni's [14] | Ours | Ni's [14] |
| 2 | 792 | 792 | 560 | 560 | 528 | 528 | 100 | 100 |
| 4 | 792 | 792 | 560 | 560 | 528 | 528 | 100 | 100 |
| 6 | 792 | 792 | 560 | 560 | 528 | 528 | 100 | 5 |
| 8 | 792 | 792 | 560 | 560 | 528 | 526 | 100 | 0 |
| 10 | 792 | 792 | 560 | 560 | 528 | 505 | 100 | 22 |

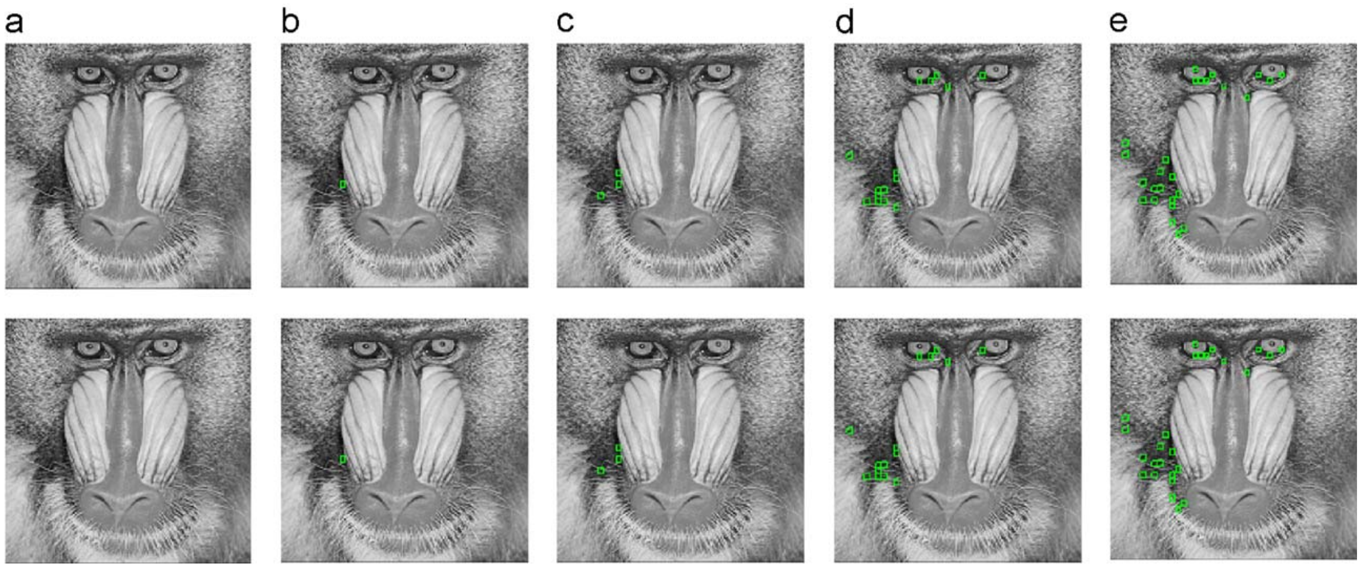


**Fig. 12.** Comparison of the locations of all unstable blocks for *Baboon*, where the upper parts are the results by Ni's method, and the lower ones are the results found by BSS. The embedding levels for (a), (b), (c), (d) and (e) are 2, 4, 6, 8 and 10, respectively.

**Table 13**
Comparison of recovery of watermark.

| Embedding level | *Cpic*1 | | *Cpic*2 | | *Cpic*3 | | *Mpic*2 | |
|---|---|---|---|---|---|---|---|---|
| | Ours | Ni's [14] | Ours | Ni's [14] | Ours | Ni's [14] | Ours | Ni's [14] |
| 2 | 1296 | 1292 | 1296 | 1296 | 763 | 763 | 100 | 100 |
| 4 | 1296 | 1296 | 1296 | 1296 | 763 | 763 | 100 | 100 |
| 6 | 1296 | 1296 | 1296 | 1296 | 763 | 763 | 100 | 100 |
| 8 | 1296 | 1296 | 1296 | 1296 | 763 | 763 | 100 | 100 |
| 10 | 1296 | 1296 | 1296 | 1296 | 763 | 763 | 100 | 100 |

**Table 14**
Comparison of PSNR (dB) between our method and method in [14].

| Embedding level | *Lena* | | *Baboon* | | *Boat* | | *Mpic*1 | |
|---|---|---|---|---|---|---|---|---|
| | Ours | Ni's [14] | Ours | Ni's [14] | Ours | Ni's [14] | Ours | Ni's [14] |
| 2 | 45.12 | 45.12 | 45.47 | 45.47 | 45.57 | 45.57 | 56.14 | 56.14 |
| 4 | 39.10 | 39.10 | 39.45 | 39.45 | 39.55 | 39.55 | 50.12 | 50.12 |
| 6 | 35.58 | 35.58 | 35.92 | 35.92 | 36.03 | 36.03 | 46.60 | 46.60 |
| 8 | 33.08 | 33.08 | 33.43 | 33.43 | 33.53 | 33.53 | 44.10 | 44.10 |
| 10 | 31.14 | 31.14 | 31.49 | 31.49 | 31.59 | 31.59 | 42.16 | 42.16 |

**Table 15**
Comparison of PSNR (dB) between our method and method in [14].

| Embedding level | *Cpic*1 | | *Cpic*2 | | *Cpic*3 | | *Mpic*2 | |
|---|---|---|---|---|---|---|---|---|
| | Ours | Ni's [14] | Ours | Ni's [14] | Ours | Ni's [14] | Ours | Ni's [14] |
| 2 | 45.75 | 45.57 | 45.78 | 45.60 | 45.86 | 45.73 | 51.42 | 51.42 |
| 4 | 39.56 | 39.55 | 39.59 | 39.58 | 39.73 | 39.71 | 50.12 | 45.40 |
| 6 | 36.04 | 36.03 | 36.07 | 36.07 | 36.21 | 36.21 | 41.87 | 41.87 |
| 8 | 33.55 | 33.55 | 33.58 | 33.58 | 33.73 | 33.73 | 39.38 | 39.38 |
| 10 | 31.62 | 31.62 | 31.69 | 31.65 | 31.81 | 31.81 | 37.44 | 37.44 |

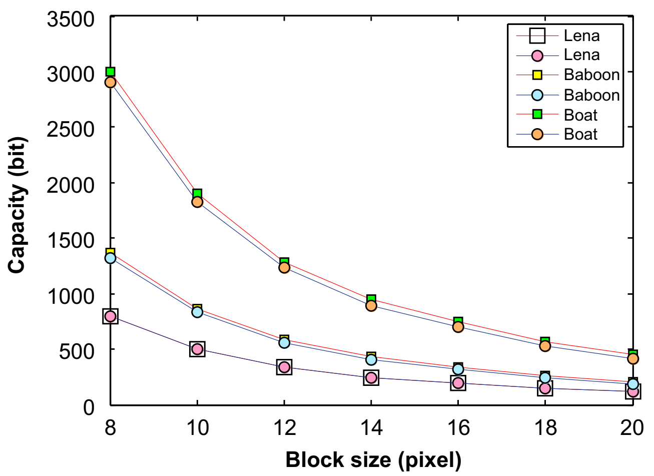


**Fig. 13.** Comparison of capacity (three magenta solid lines are associated with the proposed algorithm, three red dashed lines are with the algorithm reported in [14]), vertical axis stands for capacity in terms of bit and horizontal axis stands for block size. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

## Acknowledgements

We would like to thank the helpful comments and suggestions from the anonymous reviewers. This research was supported by National Science Foundation of China (60771068, 60702061, 60832005), the Open-End Fund of National Laboratory of Pattern Recognition in China and National Laboratory of Automatic Target Recognition, Shenzhen University, China, and the Program for Changjiang Scholars and innovative Research Team in University of China (IRT0645).

## References

[1] B.B. Zhu, M.D. Swanson, A.H. Tewfik, When seeing isn't believing, IEEE Signal Process. Mag. 21 (2) (2004) 40–49.
[2] S.R.M. Oliveira, M.A. Nascimento, O.R. Zaiane, Digital watermarking: its status, limitations and prospects, Technical Report TR-02-01, Department of Computing Science, Alberta University, Edmonton, Alberta, Canada, January 2002.
[3] J. Feng, I. Lin, C. Tsai, Y. Chu, Reversible watermarking: current status and key issues, Int. J. Network Security 2 (3) (2006) 161–171.

[4] C.W. Honsinger, P.W. Jones, M. Rabbani, J.C. Stoffel, Lossless recovery of an original image containing embedded data, US Patent, Docket No: 77102/E-D, 2001.

[5] J. Fridrich, M. Goljan, R. Du, Invertible authentication, Proc. SPIE 3971 (2001) 197–208.

[6] M.U. Celik, G. Sharma, A.M. Tekalp, E. Saber, Lossless generalized-LSB data embedding, IEEE Trans. Image Process. 14 (2) (2005) 253–266.

[7] X. Wu, N. Memon, Context-based, adaptive, lossless image coding, IEEE Trans. Commun. 45 (4) (1997) 437–444.

[8] J. Tian, Reversible data embedding using a difference expansion, IEEE Trans. Circuits Syst. Video Technol. 13 (8) (2003) 890–896.

[9] A.M. Alattar, Reversible watermark using the difference expansion of a generalized integer transform, IEEE Trans. Image Process. 13 (8) (2004) 1147–1156.

[10] L. Kamstra, H.J.A.M. Heijmans, Reversible data embedding into images using wavelet techniques and sorting, IEEE Trans. Image Process. 14 (12) (2005) 2082–2090.

[11] D.M. Thodi, J.J. Rodríguez, Expansion embedding techniques for reversible watermarking, IEEE Trans. Image Process. 16 (3) (2007) 721–730.

[12] H.J. Kim, V. Sachnev, Y. Shi, J. Nam, H. Choo, A novel difference expansion transform for reversible data embedding, IEEE Trans. Inf. Forensics Security 3 (3) (2008) 456–465.

[13] C.D. Vleeschouwer, J. Delaigle, B. Macq, Circular interpretation of bijective transformations in lossless watermarking for media asset management, IEEE Trans. Multimedia 5 (1) (2003) 97–105.

[14] Z. Ni, Y. Shi, N. Ansari, W. Su, Q. Sun, X. Lin, Robust lossless image data hiding designed for semi-fragile image authentication, IEEE Trans. Circuits Syst. Video Technol. 18 (4) (2008) 497–509.

[15] C. Deng, X. Gao, D. Tao, X. Li, Geometrically invariant watermarking using affine covariant regions, IEEE Int. Conf. Image Process. (2008) 413–416.

[16] C. Deng, X. Gao, X. Li, D. Tao, Invariant image watermarking based on local feature regions, Cyberworlds (2008) 6–11.

[17] C. Deng, X. Gao, D. Tao, X. Li, Digital watermarking in image affine co-variant regions, in: IEEE International Conference on Machine Learning and Cybernetics, vol. 4, 2007 pp. 2125–2130.

[18] D. Zou, Y. Shi, Z. Ni, W. Su, A semi-fragile lossless digital watermarking scheme based on integer wavelet transform, IEEE Trans. Circuits Syst. Video Technol. 16 (10) (2006) 1294–1300.

[19] G. Xuan, C. Yang, Y. Shi, Y. Zheng, Z. Ni, Reversible data hiding based on wavelet spread spectrum, in: Proceedings of IEEE Workshop on Multimedia Signal Processing, 2004, pp. 211–214.

[20] G. Xuan, Q. Yao, C. Yang, J. Gao, P. Chai, Y. Shi, Z. Ni, Lossless data hiding using histogram shifting method based on integer wavelets, in: Lecture Notes in Computer Science, vol. 4283, 2006, pp. 323–332.

[21] Z. Zhao, N. Yu, X. Li, A novel video watermarking scheme in compression domain based on fast motion estimation, in: IEEE International Conference on Communication Technology Proceedings, vol. 2, 2003, pp. 1878–1882.

[22] X. Li, Watermarking in secure image retrieval, Pattern Recognition Lett. 24 (14) (2003) 2431–2434.

[23] J. Hwang, J. Kim, J. Choi, A reversible watermarking based on histogram shifting, in: Lecture Notes in Computer Science, vol. 4283, 2006, pp. 348–361.

[24] W. Kuo, D. Jiang, Y. Huang, Reversible data hiding based on histogram, in: Lecture Notes in Computer Science, vol. 4682, 2007, pp. 1152–1161.

[25] D. Tao, X. Li, X. Wu, S.J. Maybank, General tensor discriminant analysis and gabor features for gait recognition, IEEE Trans. Pattern Anal. Machine Intell. 29 (10) (2007) 1700–1715.

[26] D. Tao, X. Li, W. Hu, S.J. Maybank, X. Wu, Supervised tensor learning, Knowledge and Information Systems 13 (1) (2007) 1–42.