*Intelligent Automation and Soft Computing*, Vol. 17, No. 2, pp. 233-243, 2011

# REVERSIBLE VISIBLE WATERMARKING WITH LOSSLESS DATA EMBEDDING BASED ON DIFFERENCE VALUE SHIFT

**XINPENG ZHANG, SHUOZHONG WANG AND GUORUI FENG**
*School of Communication and Information Engineering*
*Shanghai University*
*Shanghai 200072, China*
*E-mail: {xzhang, shuowang, grfeng}@shu.edu.cn*

ABSTRACT—This paper first proposes a novel lossless data-hiding method, in which the magnitudes of gray level differences in pixel pairs are slightly increased so that the region of differences with small magnitudes can be used to carry the hidden message. Overflow is avoided by prohibiting modifications to certain pixel pairs, and a small number of labels marking these exceptions are also embedded for perfect recovery of the host. Since this method avoids embedding any compressed data of location map or original data, the payload-distortion performance is very high. The lossless data-hiding method is then applied to implement a reversible visible watermarking scheme. After semi-transparently embedding a binary watermark image, some additional data about the mark and the host image are also inserted in the lossless manner. When the visible watermark is extracted from the marked version, the original host image can be completely restored without any error.

Key Words: lossless data hiding, visible watermarking, difference value shift

## 1. INTRODUCTION

As an effective means for protection of intellectual property rights, digital watermarking has been rapidly developed in recent years. While most watermarks are invisibly embedded into a host signal, a semi-transparent logo, i.e., visible watermark, directly indicates the copyright of digital contents [1,2]. In most applications, distortion introduced by watermark embedding, whether being robust or fragile, is non-removable. However, such distortion, no matter how small it is, is unacceptable in some applications, e.g., military or medical images. In this case it is imperative to embed the watermark in a lossless or invertible manner so that the original contents can be perfectly restored after extraction of the watermark [3~8]. The lossless data-hiding technique can be combined with fragile watermarking. When a digital signature of the host content is embedded as a fragile watermark using a lossless data-hiding technique, a receiver can detect any modification to the marked medium in case it has been tempered, otherwise the original host data can be retrieved free of any error.

A number of lossless embedding algorithms have been proposed. In [3], after pseudo-randomly segmenting each image block into two sub-regions, rotation of the histograms of the two sub-regions on this circle is used to embed one bit in each block. Based on this method, a robust lossless data hiding scheme is proposed [4], which can be used for semi-fragile image authentication. Payloads of the two methods are low since each block can only carry one bit. In the RS method [5], a regular-singular status is defined for each group of pixels according to a flipping operation and a discrimination function. The entirety of RS status is then losslessly compressed to provide a space for data hiding. Alternatively, the least significant digits of pixel values in an L-ary system [6] or the least significant bits of quantized DCT coefficients in a JPEG image [7] can also be used to provide the required data space. In the difference expansion (DE) algorithm [8], differences between two adjacent pixels are doubled so that a new LSB plane without carrying any information of the original is generated. The hidden message together with a compressed location map derived from the property of each pixel pair is embedded into the generated LSB plane. Since almost each pixel pair can carry one bit, the DE algorithm can embed a fairly large amount of secret data into a host image. Furthermore, various techniques have been introduced into DE algorithm to improve the payload-distortion performance, including generalized integer transform [9], histogram shift [10], prediction of location map [11], and simplification of location map [12]. In these lossless embedding techniques, a spare place can always be made available to accommodate secret information as long as the chosen item is compressible.

The paper first proposes a novel lossless data-hiding approach, in which differences in adjacent pixel pairs are shifted slightly in a direction departing from zero to create a space for data embedding. Since this approach avoids embedding any compressed data of location map or original data, most of the created space is available for the watermark itself. So the payload-distortion performance is very high. A reversible visible watermarking scheme is also proposed. A binary watermark image such as company logo is semi-transparently inserted into the host image for explicit copyright declaration. The inserted logo can be completely removed by authorized users, and the original image perfectly restored without any error.

## 2. LOSSLESS DATA-HIDING USING DIFFERENCE VALUE SHIFT

In the proposed lossless data-hiding algorithm termed *difference value shift* (DVS), the magnitudes of gray level differences in pixel pairs are slightly increased so that the gray levels of pixels are moved towards pure white or black, hence the region of difference close to zero is spared for the hidden data. By prohibiting modifications to certain pixel pairs, overflow can be avoided. Special labels marking these exceptions are also embedded for lossless recovery of the host image. This lossless data-hiding method does not involve any compression operations and, since pixel pairs unsuitable for data carrying are rare, high embedding capacity can be achieved. On the receiving side, after extraction of both the embedded secret data and the labels, the original image is recovered with the inverse DVS operation using the labels.

### *2.1 Data embedding procedure*

Assume the numbers of rows and columns of a host image are $u$ and $v$ respectively. Divide all pixels into many non-overlapping blocks, each containing two adjacent pixels. The image is thus divided into $uv/2$ pairs, and then rearranged into a sequence of pairs in a pseudo-random way derived from a secret key: $\{p_{1,1}, p_{1,2}\}$, $\{p_{2,1}, p_{2,2}\}$, …, $\{p_{u\cdot v/2,1}, p_{u\cdot v/2,2}\}$. Find the difference of each pair:

$$d_k = p_{k,1} - p_{k,2}\ , \qquad k = 1,2,\ldots,uv/2 \tag{1}$$

Assign all possible pixel value differences into three disjoint sets:

**S**$_0$: $-M \le d_k \le M-1$;

**S**$_1$: $-2M \le d_k \le -M-1$ or $M \le d_k \le 2M-1$;

**S**$_2$: $d_k \le -2M-1$ or $d_k \ge 2M$.

where the system parameter $M$ shared by the data-hider and the receiver is a positive constant, which must not be too large since, as will be clear later, it is directly associated to the extent of distortion introduced by the data embedding.

Introduce two non-negative integers $m_1$ and $m_2$ whose sum is $M$:

$$m_1 = \lfloor M/2 \rfloor, \quad m_2 = \lceil M/2 \rceil \tag{2}$$

Define the DVS (difference value shift) operation as

$$\begin{cases} p'_{k,1} = p_{k,1} - m_1, \quad p'_{k,2} = p_{k,2} + m_2, & \text{if } d_k < 0 \\ p'_{k,1} = p_{k,1} + m_2, \quad p'_{k,2} = p_{k,2} - m_1, & \text{if } d_k \geq 0 \end{cases} \tag{3}$$

which makes a larger pixel value even larger, and a smaller one even smaller. In other words, the DVS operation enlarges the magnitude of pixel differences by $M$ and pushes the pixel values towards pure white or pure black. After a DVS operation, an original $d_k$ value in $\mathbf{S}_0$ will be moved to $\mathbf{S}_1$, and an original $d_k$ value in $\mathbf{S}_1$ moved to $\mathbf{S}_2$. Anything in $\mathbf{S}_2$ will always remain in $\mathbf{S}_2$.

Assign pixel pairs, $(p_{k,1}, p_{k,2})$, to the following three groups according to its overflow property:

1) $\mathbf{T}_0$, which contains all pairs satisfying the following condition so that two consecutive DVS operations will not cause any overflow:

$$\begin{cases} 0 \leq p_{k,1} - 2m_1 \leq 255 \quad \text{and} \quad 0 \leq p_{k,2} + 2m_2 \leq 255, & \text{when } d_k < 0 \\ 0 \leq p_{k,1} + 2m_2 \leq 255 \quad \text{and} \quad 0 \leq p_{k,2} - 2m_1 \leq 255, & \text{when } d_k \geq 0 \end{cases} \tag{4}$$

2) $\mathbf{T}_1$, which contains all pairs that do not belong to $\mathbf{T}_0$ but satisfy the following condition so that only one DVS operation is permitted without causing overflow:

$$\begin{cases} 0 \leq p_{k,1} - m_1 \leq 255 \quad \text{and} \quad 0 \leq p_{k,2} + m_2 \leq 255, & \text{when } d_k < 0 \\ 0 \leq p_{k,1} + m_2 \leq 255 \quad \text{and} \quad 0 \leq p_{k,2} - m_1 \leq 255, & \text{when } d_k \geq 0 \end{cases} \tag{5}$$

3) $\mathbf{T}_2$, which contains all pairs that belong neither to $\mathbf{T}_0$ nor to $\mathbf{T}_1$:

$$\begin{cases} p_{k,1} - m_1 < 0 \quad \text{or} \quad p_{k,2} + m_2 > 255, & \text{when } d_k < 0 \\ p_{k,2} - m_1 < 0 \quad \text{or} \quad p_{k,1} + m_2 > 255, & \text{when } d_k \geq 0 \end{cases} \tag{6}$$

Obviously, $\mathbf{T}_2$ contains all the pixel pairs in which the DVS operation is prohibited, and any pixel pair in group $\mathbf{T}_1$ will be moved to group $\mathbf{T}_2$ after one DVS operation.

We will shift the differences in $\mathbf{S}_1$ and $\mathbf{S}_2$ towards the positive or negative infinite directions to empty set $\mathbf{S}_1$ and, after mapping each difference value in $\mathbf{S}_0$ to a bit to be embedded, to move the difference values into $\mathbf{S}_1$ if the mapped bits are ones or to keep them unchanged if the mapped bits are zeros. In other words, the two sets $\mathbf{S}_0$ and $\mathbf{S}_1$ are used to represent embedded data 0 and 1, respectively. In order to prevent any possible overflow caused by DVS operations and ensure perfect recovery of the original image, the pixel pairs belonging to $\mathbf{T}_1$ or $\mathbf{T}_2$, meaning that the gray levels are close to pure white/black, must be carefully treated. A few bits are used to label whether DVS operation should be performed to these special pairs. Both the secret message and the additional bits are then embedded into the pixel pair belonging to $\mathbf{T}_0$ and having a difference in $\mathbf{S}_0$.

A pixel pair belonging to $\mathbf{T}_0$ and having a difference in $\mathbf{S}_0$ is used to carry one bit, while no change should be made to a pair that does not belong to $\mathbf{T}_0$ if the difference falls in $\mathbf{S}_0$. On the other hand, any pair with a pixel difference in $\mathbf{S}_1$ or $\mathbf{S}_2$ is modified with a DVS operation as long as it does not belong to $\mathbf{T}_2$, and the resulting pair always has a difference in $\mathbf{S}_2$. Thus, $\mathbf{S}_1$ is spared to accommodate a possible shift from $\mathbf{S}_0$ when data embedding takes place in certain circumstance. To avoid overflow, a pair belonging to $\mathbf{T}_2$ should never be altered. In order to recover the host content after data extraction, the receiver must be able to distinguish between two cases: a pixel pair originally belonging to $\mathbf{T}_2$, and a pixel pair originally belonging to $\mathbf{T}_1$ but being moved to $\mathbf{T}_2$ as a result of a DVS operation. A set of labels is therefore introduced in the scheme, and should be embedded into the host image together with the secret message. In case of an $\mathbf{S}_0/\mathbf{T}_0$ combination, if the bit to be embedded is 0, the original pixel values in the pair are kept unchanged, otherwise a DVS operation is performed to push the difference to region $\mathbf{S}_1$. In other words, an embedded 0 or 1 can be identified, respectively, by the magnitude of difference of the modified pair as being in $\mathbf{S}_0$ or $\mathbf{S}_1$.

The complete procedure of data embedding is as follows.

1) if $d_k \in \mathbf{S}_2$

If $(p_{k,1}, p_{k,2}) \in \mathbf{T}_2$, label the pair with ‘1’ and keep the original gray levels;

If $(p_{k,1}, p_{k,2}) \in \mathbf{T}_1$, label the pair with ‘0’ and shift the difference value according to (3), leading to $d'_k \in \mathbf{S}_2$ and $(p'_{k,1}, p'_{k,2}) \in \mathbf{T}_2$;

If $(p_{k,1}, p_{k,2}) \in \mathbf{T}_0$, shift the difference value according to (3), leading to $d'_k \in \mathbf{S}_2$ and $(p'_{k,1}, p'_{k,2}) \in \mathbf{T}_0$ or $\mathbf{T}_1$;

2) if $d_k \in \mathbf{S}_1$

If $(p_{k,1}, p_{k,2}) \in \mathbf{T}_2$, keep the original gray levels;

If $(p_{k,1}, p_{k,2}) \in \mathbf{T}_1$, label the pair with ‘0’ and shift the difference value according to (3), leading to $d'_k \in \mathbf{S}_2$ and $(p'_{k,1}, p'_{k,2}) \in \mathbf{T}_2$;

If $(p_{k,1}, p_{k,2}) \in \mathbf{T}_0$, shift the difference value according to (3), leading to $d'_k \in \mathbf{S}_2$ and $(p'_{k,1}, p'_{k,2}) \in \mathbf{T}_0$ or $\mathbf{T}_1$;

3) if $d_k \in \mathbf{S}_0$

If $(p_{k,1}, p_{k,2}) \in \mathbf{T}_2$, keep the original gray levels;

If $(p_{k,1}, p_{k,2}) \in \mathbf{T}_1$, keep the original gray levels;

If $(p_{k,1}, p_{k,2}) \in \mathbf{T}_0$, embed one bit in each pixel pair. The embedded data include all labels and the additional message. When the embedded bit is 0, keep the original gray levels, otherwise the difference value is shifted according to (3), leading to $d'_k \in \mathbf{S}_1$ and $(p'_{k,1}, p'_{k,2}) \in \mathbf{T}_0$ or $\mathbf{T}_1$.

A necessary condition for the proposed DVS method to be applicable is that the number of pixel pairs that belong to $\mathbf{T}_0$ and possess a difference in $\mathbf{S}_0$ must be greater than the number of labels, which is a sum of three numbers: the number of pixel pairs that belong to $\mathbf{T}_2$ with a $d_k$ in $\mathbf{S}_2$, the number of pixel pairs that belong to $\mathbf{T}_1$ with a $d_k$ in $\mathbf{S}_2$, and the number of pixel pairs that belong to $\mathbf{T}_1$ with a $d_k$ in $\mathbf{S}_1$. This ensures availability of a spare space for the hidden data. As the values of adjacent pixels are usually very close and the number of pixels with gray levels near saturation is small, most differences are in $\mathbf{S}_0$ and only a few pixel pairs belong to $\mathbf{T}_2$ or $\mathbf{T}_1$, indicating that the proposed scheme is practical.

### *2.2 Data extraction and image restoration*

With the secret key used in data embedding, the pattern of pixel assignment into $uv/2$ pairs can be retrieved at the receiver. The same $\mathbf{T}_0$, $\mathbf{T}_1$ and $\mathbf{T}_2$ are used to classify the received pairs, and $\mathbf{S}_0$, $\mathbf{S}_1$ and $\mathbf{S}_2$ to classify the received differences of pixel pairs, $d'_k$s. The embedded message and the labels to be used for the host image recovery can be separated from the data extracted from differences in $\mathbf{S}_0$ and $\mathbf{S}_1$, since the number of labels equals the number of pixel pairs belonging to $\mathbf{T}_2$ and having a difference in $\mathbf{S}_2$. Using the labels and inverse DVS operations, the original image can be recovered without any error.

Denoting the received pixel pair as $(p'_{k,1}, p'_{k,2})$, an inverse DVS operation can be defined:

$$\begin{cases} p_{k,1} = p'_{k,1} + m_1, \quad p_{k,2} = p'_{k,2} - m_2, & \text{if } d_k < 0 \\ p_{k,1} = p'_{k,1} - m_2, \quad p_{k,2} = p'_{k,2} + m_1, & \text{if } d_k \geq 0 \end{cases} \tag{7}$$

The following gives the complete procedure of data extraction and host image recovery:

1) if $d'_k \in \mathbf{S}_0$

If $(p'_{k,1}, p'_{k,2}) \in \mathbf{T}_0$, keep the received gray levels and extract an embedded ‘0’;

If $(p'_{k,1}, p'_{k,2}) \in \mathbf{T}_1$, keep the received gray levels;

If $(p'_{k,1}, p'_{k,2}) \in \mathbf{T}_2$, keep the received gray levels;

2) if $d'_k \in \mathbf{S}_1$

If $(p'_{k,1}, p'_{k,2}) \in \mathbf{T}_0$, perform an inverse DVS to recover the original gray levels and extract an embedded ‘1’;

If $(p'_{k,1}, p'_{k,2}) \in \mathbf{T}_1$, perform an inverse DVS to recover the original gray levels and extract an embedded ‘1’;

If $(p'_{k,1}, p'_{k,2}) \in \mathbf{T}_2$, keep the received gray levels;

3) if $d'_k \in \mathbf{S}_2$

If $(\mathrm{p}'_{\mathrm{k},1}, p'_{k,2}) \in \mathbf{T}_0$, perform an inverse DVS to recover the original gray levels;

If $(\mathrm{p}'_{\mathrm{k},1}, p'_{k,2}) \in \mathbf{T}_1$, perform an inverse DVS to recover the original gray levels;

If $(p'_{k,1}, p'_{k,2}) \in \mathbf{T}_2$, recover the original gray levels according to the extracted labels. The extracted data are divided into two parts as mentioned above: the hidden message, and the labels corresponding to the pixel pairs that belong to $\mathbf{T}_2$ and possess a difference in $\mathbf{S}_2$. If a pixel pair is labeled ‘1’, an inverse DVS is performed to recover the original gray levels, and the gray values in received pairs labeled ‘0’ are just the original contents.

As has been stated, the number of embedded labels is equal to the total number of pixel pairs that have been moved from the status $\mathbf{S}_2/\mathbf{T}_2$, $\mathbf{S}_2/\mathbf{T}_1$ and $\mathbf{S}_1/\mathbf{T}_1$ to $\mathbf{S}_2/\mathbf{T}_2$.

The methods of data embedding, extraction, and host recovery are summarized in Table I.

**Table I. Data embedding, extraction and host recovery procedures**

| Original attributes | | Data embedding | | New attributes | | Data extraction and host recovery | |
|---|---|---|---|---|---|---|---|
| | | DVS | Label | | | Inverse DVS | Extracted bits |
| $\mathbf{S}_2$ | $\mathbf{T}_2$ | No | 1 | $\mathbf{S}_2$ | $\mathbf{T}_2$ | No, if labeled by ‘1’<br>Yes, if labeled by ‘0’ | — |
| | $\mathbf{T}_1$ | Yes | 0 | | | | — |
| | $\mathbf{T}_0$ | Yes | — | | $\mathbf{T}_0$ or $\mathbf{T}_1$ | Yes | — |
| $\mathbf{S}_1$ | $\mathbf{T}_2$ | No | — | $\mathbf{S}_1$ | $\mathbf{T}_2$ | No | — |
| | $\mathbf{T}_1$ | Yes | 0 | $\mathbf{S}_2$ | $\mathbf{T}_2$ | Yes (the label is ‘0’) | — |
| | $\mathbf{T}_0$ | Yes | — | | $\mathbf{T}_0$ or $\mathbf{T}_1$ | Yes | — |
| $\mathbf{S}_0$ | $\mathbf{T}_2$ | No | — | $\mathbf{S}_0$ | $\mathbf{T}_2$ | No | — |
| | $\mathbf{T}_1$ | No | — | | $\mathbf{T}_1$ | No | — |
| | $\mathbf{T}_0$ | No, if a ‘0’ is embedded | — | | $\mathbf{T}_0$ | No | 0 |
| | | Yes, if a ‘1’ is embedded | | $\mathbf{S}_1$ | $\mathbf{T}_0$ or $\mathbf{T}_1$ | Yes | 1 |

## 3. DVS PERFORMANCE ANALYSIS

In the following, we study the relationship between data hiding induced distortion and the payload. The pixel pairs belonging to $\mathbf{T}_1$ and $\mathbf{T}_2$ are ignored in the discussion since their number is generally very small. As has been mentioned, all pixel pairs belonging to $\mathbf{T}_0$ and having a difference in $\mathbf{S}_0$ are used to carry data. The larger the parameter $M$ is assigned, the larger the set $\mathbf{S}_0$ and therefore the greater the actual payload. Assume that 0 and 1 in the secret message are uniformly distributed. As a result of data embedding, 50% of the pixel pairs in average are kept unchanged, which are used to represent the embedded 0, and the other 50% undergo the DVS operation, representing the embedded 1. Denoting the ratio between the number of pixel pairs belonging to $\mathbf{T}_0$ and having a difference in $\mathbf{S}_0$ and the number of all pairs as $P_{00}$, the energy of distortion caused by DVS embedding is

$$D_{\mathrm{E}} = \left(1 - \frac{P_{00}}{2}\right) \cdot \left(\frac{m_1^2}{2} + \frac{m_2^2}{2}\right) \cdot u \cdot v \tag{8}$$

PSNR due to data embedding is therefore obtained:

$$\mathrm{PSNR} = -10 \cdot \log_{10} \frac{(1 - P_{00}/2) \cdot \left(m_1^2/2 + m_2^2/2\right)}{255^2} \tag{9}$$

Four images, Lena, Baboon, Lake, and Barbara, all sized 512×512, were used as the host media. Define a parameter $D_{\mathrm{A}}$ to represent the average energy of the distortion caused by embedding one bit. Figure 1 shows $D_{\mathrm{A}}$ computed for different host images with several values of $M$ represented by different symbols. The ordinate represents $D_{\mathrm{A}}$ values, and the abscissa is the net payload in bits per pixel, i.e., the ratio between the length of embedded bit sequence and size of the host image. Generally speaking, a smaller $M$ gives lower distortion and less net payload.

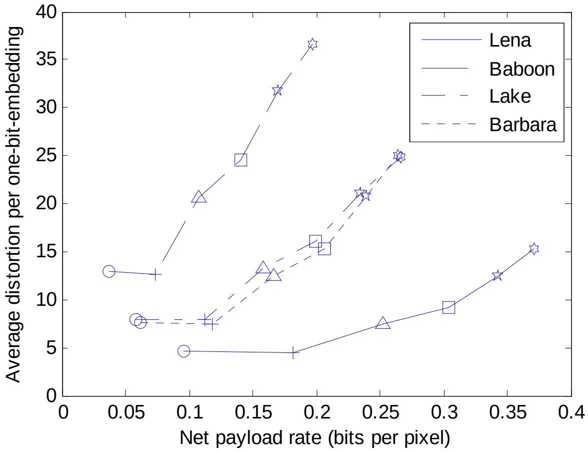
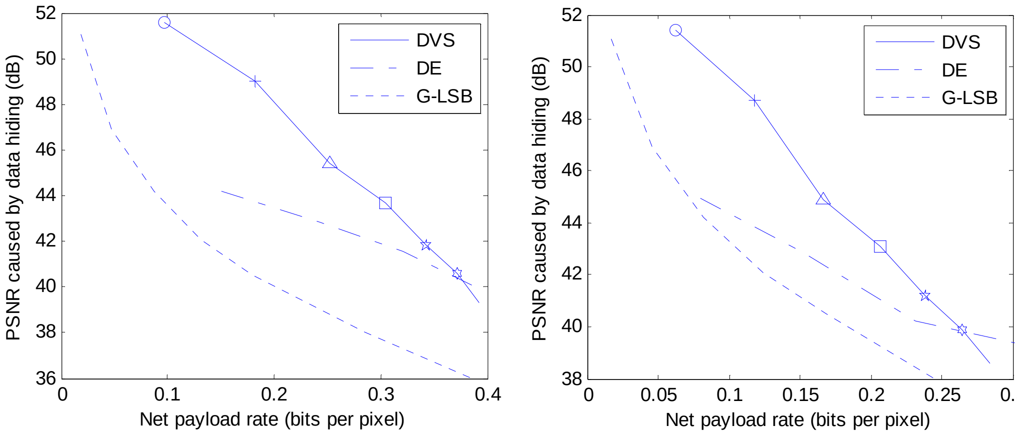


**Figure 1. Performance of the proposed lossless data hiding scheme at different *M* values (circles: *M* = 1; crosses: *M* = 2; triangles: *M* = 3; squares: *M* = 4; pentagrams: *M* = 5; hexagrams: *M* = 6)**

For lossless data-hiding methods, the more the data are embedded, the more distortion is caused. Figure 2 presents a performance comparison of three methods, G-LSB [6], DE [8] and the proposed DVS, using Lena and Barbara as the host. The DVS technique provides better payload-distortion performance than the G-LSB method at any embedding payload tested in the experiment, and is better than the DE method when the embedding rate is relatively low, corresponding to high PSNR situations that is considered desirable for many applications requiring good imperceptibility. Experiments on other images provide similar results.



**(a)** **(b)**

**Figure 2. Performance comparison of the three methods: DE, G-LSB, and the proposed DVS. (a) Lena; (b) Barbara**

## 4. VISIBLE WATERMARKING BASED ON DVS

A DVS-based visible watermarking scheme is also proposed, in which a binary mark image is semi-transparently embedded into a host image, and some additional information about the watermark and host image to be used for the original content recovery are also embedded with the DVS algorithm. The receiver is able to decide whether the visibly watermarked image has been tampered, and, if not, to extract the embedded binary watermark and completely restore the original host image without any error.

Assume the size of a host image is $u_H \times v_H$, and that of a binary watermark image $u_w \times v_w$. Generally, $u_w < u_H$ and $v_w < v_H$. Periodically repeat the mark image to make the size equal to $u_H \times v_H$. Denoting gray levels of the host image as $g(i, j)$ and those of the periodically extended watermark as $w(i, j)$. Calculate

$$g'(i,j) = \begin{cases} g(i,j)+C, & \text{if } w(i,j)=1 \text{ and } g(i,j) \le 255-C \\ g(i,j), & \text{if } w(i,j)=1 \text{ and } g(i,j) > 255-C \\ g(i,j)-C, & \text{if } w(i,j)=0 \text{ and } g(i,j) \ge C \\ g(i,j), & \text{if } w(i,j)=0 \text{ and } g(i,j) < C \end{cases} \tag{10}$$

where $C$ is a parameter selected by the watermark hider. A larger $C$ corresponds to a stronger visible watermark. The value of $C$ may be selected according to the desired visual effect. To avoid any overflow, pixels with gray levels near pure white/black are unaltered. Denote the image with a visible watermark as **G'**.

Apart from the visible watermark, additional data consisting of the following four parts should also be embedded.

i) A hash of **G'** obtained from (19), $A_1 = h(\mathbf{G'})$. $A_1$ will be used on the receiving side to check whether a received image has been modified during transmission, and to find the embedding position of the additional data.

ii) The losslessly compressed original (unrepeated) watermark $A_2$.

iii) A binary representation of the value $C$, $A_3$.

iv) A set of binary tags, $A_4$, used to label the pixels with gray values near pure white/black. A ‘0’ is used to label each pixel meeting $g(i, j) > 255–C$ and $w(i, j) = 1$ or meeting $g(i, j) < C$ and $w(i, j) = 0$, which are unaltered in visible watermark embedding. A ‘1’ is used to label each pixel meeting $255–2C < g(i, j) \le 255–C$ and $w(i, j) = 1$ or meeting $C \le g(i, j) < 2C$ and $w(i, j) = 0$, which has been respectively decreased or increased by $C$ according to (19). This means that the values of all labeled pixels in **G'** are less than $C$ or greater than $255–C$.

Concatenate $A_1$, $A_2$, $A_3$ and $A_4$ to produce the additional data **A**, and denote the number of bits in **A** as $L_{\mathbf{A}}$.

Use the DVS algorithm to embed the additional data **A** into **G'** to produce the final watermarked image **G''**. As a larger *M* corresponds to a higher payload with greater distortion, we seek the smallest possible *M* value that can provide sufficient space to accommodate all bits in **A**, and perform a DVS embedding. It is unnecessary to transmit the value of *M* since it can be derived form the watermarked image at the receiver. Here, a secret key is used to determine the order in accessing pixel pairs and the order in accessing pixels of each pair. Any unauthorized user cannot restore the host image without this key.

With a watermarked image and the embedding key, the receiver can perform watermark extraction and host image restoration. Since the value of *M* used in the DVS embedding is not transmitted, the receiver must examine all possible *M* values to perform message extraction and image restoration with the DVS algorithm. If a particular value of *M* is found to make the hash of the restored image identical to the first part of the extracted message, $A_1$, this value is the one used in the embedding. On the other hand, if none of the tested *M* values meets the condition, it can be sure that the received image has been tampered. Having found the *M* value, the image **G'** and the additionally embedded **A** are recovered. From **A**, the binary watermark, the value of *C*, and the tags can be obtained. After the same periodic repetition of the mark image to make it equal the host in size, and mapping each tag into a pixel in **G'** with gray level less than *C* or more than $255–C$, the original image can be recovered as follows.

$$g(i, j) = \begin{cases} g'(i, j) - C, & \text{if } w(i, j) = 1 \text{ and } C \le g'(i, j) \le 255 - C \\ g'(i, j) + C, & \text{if } w(i, j) = 0 \text{ and } C \le g'(i, j) \le 255 - C \\ g'(i, j), & \text{if } w(i, j) = 1 \text{ and the correspond ing tag is } 0 \\ g'(i, j), & \text{if } w(i, j) = 0 \text{ and the correspond ing tag is } 0 \\ g'(i, j) - C, & \text{if } w(i, j) = 1 \text{ and the correspond ing tag is } 1 \\ g'(i, j) + C, & \text{if } w(i, j) = 0 \text{ and the correspond ing tag is } 1 \end{cases} \tag{11}$$

Two binary watermark images sized 200×200 and 100×200, as shown in Figure 12, were embedded into 512×512 host images Lena and Lake with $C$=10 and $C$=20, respectively. Images containing only visible watermarks, **G'**, are shown in Figure 13. After embedding the additional data **A** into these two images using the DVS algorithm, the output, **G''** (not shown), are visually indistinguishable from **G'**. PSNR values of **G''** with respect to **G'** are 51.7 dB and 41.6 dB for Lena and Lake, respectively, indicating that the additional data caused very little distortion. The original host images and the embedded binary watermarks can be perfectly separated using the proposed technique.

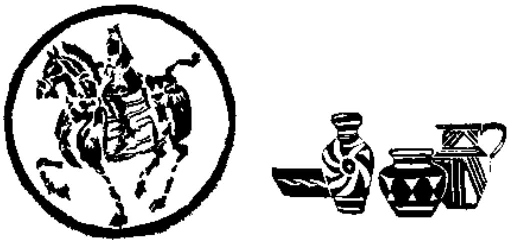


**Figure 3. Two binary watermark images**

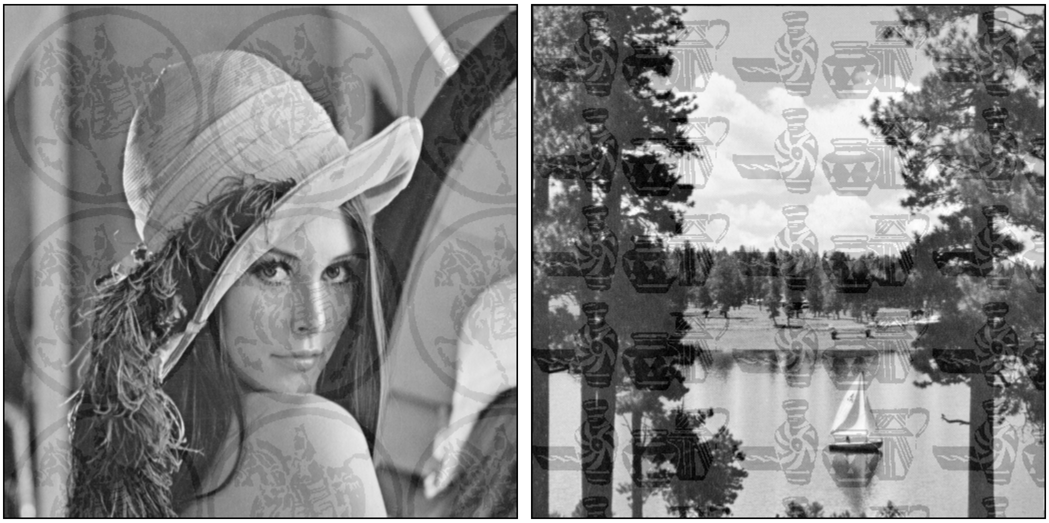


**Figure 4. Images containing visible watermarks, which are visibly identical to that with additionally embedded data A for lossless restoration of the originals**

## 5. CONCLUSION

In the proposed lossless DVS data-hiding algorithm, differences in adjacent pixel pairs are shifted slightly in a direction departing from zero to create a space for data embedding. Redundancy existing in the pixel pairs makes possible to create a sufficient spare space so that a fairly large amount of data may be embedded. Furthermore, as the extreme white/black pixels are rare, only a very small number of additional bits apart from the actual watermark are required to be embedded into the host image for restoration of the original image, therefore most of the created space is available for the watermark itself. Since the shift in magnitude is tiny, the payload-distortion performance of the DVS algorithm is excellent.

In addition to information-hiding applications where the host media are required to be perfectly restored after extraction of the embedded secret data, the DVS technique can be also used to construct a reversible visible watermarking scheme. In the scheme proposed in this paper, after a visible watermark is embedded, some additional information about the watermark and the host image, to be used for host image recovery, are also losslessly embedded. With a known key, the receiver can first decide whether a visibly watermarked image has been tampered, and, if not, extract the embedded binary watermark and completely restore the original host image.

## ACKNOWLEDGEMENT

This work was supported by the Natural Science Foundation of China (60872116, 60832010 and 60773079), the High-Tech Research and Development Program of China (2007AA01Z477), and the Natural Science Foundation of Shanghai (09ZR1412400).

## REFERENCES

1. Y. Hu, and S. Kwong, “Wavelet Domain Adaptive Visible Watermarking,” *Electronics Letters*, **37**, pp. 1219-1220, 2001.
2. S. D. Lin and S.-C. Shie, “Improving Robustness of Visible Watermarking Schemes for Images,” *Proceedings of IEEE International Symposium on Consumer Electronics*, pp. 11-14, 2004.
3. C. Vleeschouwer, J.-F. Delaigle, and B. Macq, “Circular Interpretation of Bijective Transformations in Lossless Watermarking for Media Asset Management,” *IEEE Trans. on Multimedia*, **5**(1), pp. 97-105, 2003.
4. Z. Ni, Y. Q. Shi, N. Ansari, W. Su, Q. Sun, and X. Lin, “Robust Lossless Image Data Hiding Designed for Semi-Fragile Image Authentication,” *IEEE Trans. on Circuits and Systems for Video Technology*, **18**(4), pp. 497-509, 2008.
5. M. Goljan, J. Fridrich, and R. Du, “Distortion-Free Data Embedding,” *4th International Workshop on Information Hiding, Lecture Notes in Computer Science*, **2137**, Springer-Verlag, pp.27-41, 2001.
6. M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, “Lossless Generalized-LSB Data Embedding,” *IEEE Trans. on Image Processing*, **14**(2), pp. 253-266, 2005.
7. J. Fridrich, M. Goljan, and R. Du, “Lossless Data Embedding for All Image Formats,” *Security and Watermarking of Multimedia Contents IV, Proceedings of SPIE*, **4675**, pp. 572-583, San Jose, USA, 2002.
8. J. Tian, “Reversible Data Embedding Using a Difference Expansion,” *IEEE Trans. on Circuits and Systems for Video Technology*, **13**(8), pp. 890-896, 2003.
9. A. M. Alattar, “Reversible Watermark Using the Difference Expansion of a Generalized Integer Transform,” *IEEE Trans. on Image Processing*, **13**(8), pp. 1147-1156, 2004.
10. D. M. Thodi and J. J. Rodríguez, “Expansion Embedding Techniques for Reversible Watermarking,” *IEEE Trans. on Image Processing*, **16**(3), pp. 721-730, 2007.
11. L. Kamstra, and H. J. A. M. Heijmans, “Reversible Data Embedding Into Images Using Wavelet Techniques and Sorting,” *IEEE Trans. on Image Processing*, **14**(12), pp. 2082-2090, 2005.
12. H. J. Kim, V. Sachnev, Y. Q. Shi, J. Nam, and H.-G. Choo, “A Novel Difference Expansion Transform for Reversible Data Embedding,” *IEEE Trans. on Information Forensics and Security*, **3**(3), pp. 456–465, 2008.

## ABOUT THE AUTHORS

**X. Zhang** received the B.S. degree in computational mathematics from Jilin University, China, in 1995, and the M.E. and Ph.D. degrees in communication and information system from Shanghai University, China, in 2001 and 2004, respectively. Since 2004, he has been with the faculty of the School of Communication and Information Engineering, Shanghai University, where he is currently a Professor. His research interests include information hiding, image processing and digital forensics.

**S. Wang** received B.S. degree in 1966 from Peking University, P.R. China, and Ph.D. degree in 1982 from University of Birmingham, England. He was with Institute of Acoustics, Chinese Academy of Sciences, from January 1983 to October 1985 as research fellow, and joined Shanghai University of Technology in October 1985 as associate professor. He is now professor of School of Communication and Information Engineering, Shanghai University. He was associate scientist at Department of EECS, University of Michigan, USA, from March 1993 to August 1994, and a research fellow at Department of Information Systems, City University of Hong Kong, in 1998 and 2002. His research interests include underwater acoustics, image processing, and multimedia security. He has published more than 150 papers in these areas. Many of his research projects are supported by the Natural Science Foundation of China.

**G. Feng** received the B.S. and M. S. degree in computational mathematics from Jilin University, China, in 1998 and 2001 respectively. He received Ph. D. degree in electronic engineering from Shanghai Jiaotong University, China, 2005. From January 2006 to December 2006, he was an assistant professor in East China Normal University, China. During 2007, he was a research fellow in Nanyang Technological University, Singapore. Now he is with the School of Communication and Information Engineering, Shanghai University, China. His current research interests include image processing, hiding information and computational intelligence.