Journal of Information Hiding and Multimedia Signal Processing ©2013 ISSN 2073-4212
Ubiquitous International Volume **4**, Number **2**, April **2013**

Invariability of Mean Value Based Reversible Watermarking

Shaowei Weng

Information Engineering College
Guangdong University of Technology, P. R. China

Shu-Chuan Chu

School of Computer Science, Engineering and Mathematics
Flinders University of South Australia

Nian Cai

Information Engineering College
Guangdong University of Technology, P. R. China

Rongxin Zhan

Information Engineering College, Zhengzhou University

Received December, 2012; revised January, 2013

Abstract. *By employing invariant relation between the mean value of the first $(n-1)$ pixels and the last one pixel (also called the remaining pixel) for every image block containing n pixels, a new reversible watermark scheme capable of mostly carrying $2n-3$ bits into one n-sized image block in a single embedding process is presented in this paper. First, the mean value of the first $(n-1)$ pixel is calculated. Next, the difference value between the last one pixel and this obtained mean value is applied to distinguish which classification (i.e., smooth or complex sub-block) any sub-block belongs to. Consequently, it is determined to embed $(n-2)$ bits or $2(n-2)$ bits into each sub-block according to its final classification results. And meanwhile, the mean value is reapplied to predict this last one pixel. 1-bit watermark is embedded into this last one pixel in accordance with the magnitude of prediction-error value. By multi-employing invariability of the mean value of $(n-1)$ pixels, the embedding rate can approach to $(2-\frac{3}{n})$ bpp (bit per pixel) for a single embedding process. Meanwhile, the embedding distortion is greatly controlled by embedding more bits into smooth image blocks and fewer bits into the other blocks with complex texture. Experimental results reveal the proposed scheme is effective.*

Keywords: Reversible Watermarking, Invariability of Mean Value

1. Introduction.

Reversible watermarking based on difference expansion was proposed by Tian [1]. The differences between two pixels were expanded to carry watermark information if neither overflow nor underflow occurred. Alattar [2] generalized the DE technique by taking a set containing multiple pixels rather than a pair. Coltuc *et al.* propose a threshold-controlled embedding scheme based on an integer transform for pairs of pixels [3]. In Thodi's work [4], histogram shifting was incorporated into Tian's method to produce a new algorithm called Alg. D2 with a overflow map. Although Alg. D2 has achieved high performance relative to Tian's method, its embedding rates still can not exceed 0.5 bpp (bit per pixel) for a single embedding processing. According to the performance comparison figure in paper [4], a inflexion point appeared at around 0.5 when multiple embedding was applied to achieve rates close to 1 bpp. Weng *et al.* [5] proposed an integer transform based on invariability of the sum of pixel pairs [5]. Though Weng *et al.*'s method can achieve embedding rates of about 1 bpp for a single embedding process, it is not capable of embedding small payloads at low distortions for some medical images. Weng *et al.*'s method [6] proposed a new integer transform, and the embedding rate can approach to 1 bpp for a single embedding process by overlapped pairing. In Wang *et al.*'s method [7], an generalized integer transform and a payload-dependent location map were constructed to extend the DE technique to the pixel blocks of arbitrary length.

In Wang *et al.*'s method, the variation of any image block is proposed to determine whether this block can be used for embedding or not. The variation is only a coarse estimation for correlation among pixels, which can not reflect on the detailed relationship between each pixel and its mean value. With the threshold increased, sub-blocks with weak inter-pixel correlation are also possibly classified into the embeddable sub-block, and hence, the distortion introduced by modifying these sub-blocks also is higher. In the simulate study for Wang *et al.* 's method, it is found that the distortion has a drastic increment as the embedding rate increases.

To solve this problem above, in the proposed method, all image blocks are classified into one of two categories (namely smooth and complex (also called high-textured) sub-block) according to inter-pixel correlation strength. The strength of correlation is measured depending on the relation between the mean value of the first $(n-1)$ pixels and the last one pixel. Different embedding strategy is adopted aiming at different types of blocks. In other words, more bits are embedded into smooth blocks while fewer in high-textured blocks. Histogram shifting technique is applied to the proposed method so as to more efficiently compress the location map which is used for recording the overflow or underflow positions. In fact, the introduction of histogram shifting is bound to increase distortion while have no help to an increase in payload. Therefore, at low embedding rates, the distortion introduced by histogram shifting itself is more obvious than that created by the process of embedding watermark. As a result, Wang *et al.*'s method performs better than the proposed method at low embedding rates. With the embedding bit-rates increased, the number of sub-blocks used for histogram shifting is decreased, and hence, the distortion introduced by histogram shifting is decreased. As a a result, the proposed method is significantly better than Wang *et al.* 's method at high embedding rates.

2. Wang's method.

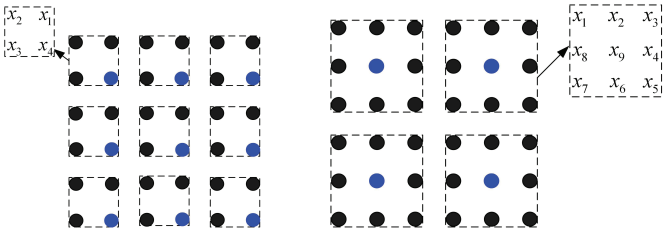
In Wang's *et al.*'s method, a generalized integer transform based on invariability of the variation of an image block is proposed.

$$\begin{aligned} y_1 &= 2x_1 - 2f(a(\mathbf{x})) + w_1 \\ &\cdots \\ y_{n-1} &= 2x_{n-1} - 2f(a(\mathbf{x})) + w_{n-1} \\ y_n &= 2x_n - a(\mathbf{x}) \end{aligned} \tag{1}$$

The integer transform defined in Wang *et al.*'s method is modified as follows.

$$\begin{aligned} y_1 &= 2x_1 - (a(\mathbf{x}) + LSB(a(\mathbf{x}))) + w_1 \\ &\cdots \\ y_{n-1} &= 2x_{n-1} - (a(\mathbf{x}) + LSB(a(\mathbf{x}))) + w_{n-1} \\ y_n &= 2x_n - a(\mathbf{x}) \end{aligned} \tag{2}$$

where $\mathbf{x}$ is a pixel block containing n pixels, i.e., $\mathbf{x} = (x_1, \cdots, x_n) \in \mathbb{Z}^n$, $a(\mathbf{x}) = \begin{cases} \lfloor \bar{\mathbf{x}} \rfloor & \text{if } \bar{\mathbf{x}} - \lfloor \bar{\mathbf{x}} \rfloor < 0.5 \\ \lceil \bar{\mathbf{x}} \rceil & \text{otherwise} \end{cases}$, $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} x_i$, $f(\cdot) = \lfloor \frac{\mathbf{x}}{2} \rfloor = \frac{\mathbf{x}+LSB(\mathbf{x})}{2}$, $LSB(\cdot)$ represents the least significant bit (LSB). $a(\mathbf{x})$ is actually the rounded value of $\bar{\mathbf{x}}$. $\mathbf{y} = (y_1, \cdots, y_n)$ is used to denote the corresponding watermarked pixel block of $\mathbf{x}$.



(a) 2 × 2-sized image block

(b) 3 × 3-sized image block

Figure 1. The corresponding partition way for a 2 × 2 and 3 × 3 images block, where $(n-1)$ pixels of any block are marked as black-filled circles, while the remaining pixel marked as the blue-filled circle

In Wang *et al.*'s method, the variation of any image block is proposed to determine whether this block can be used for embedding or not. Take an image block $\mathbf{x_1} = (13, 5, 6)$ for example, its mean value, i.e., $a(\mathbf{x_1})$, is 8, and its variation value denoted by $v(\mathbf{x}_1)$ is calculated using the following formula: $\sqrt{(x_1 - a(\mathbf{x_1}))^2 + (x_2 - a(\mathbf{x_1}))^2 + (x_2 - a(\mathbf{x_1}))^2} = \sqrt{38}$. If 2-bit watermark is set to '10', then $\mathbf{y_1} = (19, 2, 4)$ after watermark embedding. For another image block $\mathbf{x_2} = (13, 6, 8)$, its $a(\mathbf{x_2})$ is 9, and its the variation is $\sqrt{26}$. After 2-bit watermark '10' are embedded into $\mathbf{x_2}$, $\mathbf{y_2} = (18, 3, 7)$. For $\mathbf{x_1}$, its $(n-1)$ difference values which are calculated by subtracting the grayscale value of each pixel from its mean value are 5, -3, and -2 respectively. Similarly, for $\mathbf{x_2}$, the difference values are 4,-3, and -1, respectively. Here, let the given threshold denoted by the notation T_h be 7. Since $0 \leq \mathbf{y_i} \leq 255$, and $|v(\mathbf{x}_i)| \leq T_h$, two bits must be embedded into block $\mathbf{x_i}$ despite the fact that some difference value is a larger value, e.g., 5, 4 and -3, where $1 \leq i \leq 2$ and $i \in \mathbb{Z}$, and $| \cdot |$ represents the absolute value. It can be seen from the example above that the variation is only a coarse estimation for correlation among pixels, which can not reflect on the detailed relationship between each pixel and its mean value. With the threshold increased, sub-blocks with weak inter-pixel correlation are also possibly changed into the embeddable sub-block, and hence, the distortion introduced by modifying these sub-blocks also is higher. In the simulate study for Wang *et al.* 's method, it is found that the distortion has a drastic increment as the embedding rate increases.

To solve the problem above, in the proposed method, all the image blocks will firstly be classified into smooth and complex blocks according to the correlation between the mean value of the first $(n-1)$ pixels and the last one pixel. In accordance with the classification results of sub-blocks, the correlation among pixels must be reconsidered, if the correlation is strong, 2-bit watermark is embedded into this difference values of smooth blocks, and meanwhile, one bit is embedded into this difference value of complex blocks. Hence, the embedding distortion is greatly decreased by embedding more bits into smooth image blocks while embedding fewer bits into blocks with complex texture.

3. **The proposed method.** In the proposed method, a grayscale image is partitioned into non-overlapping n-sized sub-blocks, where $n = m \times m$. Any $m \times m$-sized sub-block $\mathbf{x}_n$ is arranged into a dimensional list based on high-correlation existing between two neighboring pixels, i.e., $\mathbf{x}_n = (x_1, x_2, \cdots, x_n) \in \mathbb{Z}^n$, its first $(n-1)$ pixels are grouped into a sub-set denoted by $\mathbf{x_{n-1}}$.

Take a 2 × 2 sub-block for example, the bottom-right pixel x_4 marked as blue-filled circle is selected as the remaining pixel, the other three pixels are selected to form sub-set $\mathbf{x_{n-1}}$ as illustrated in Fig. 1(a). The center pixel x_9 of a 3 × 3 image block is selected to be this remaining pixel, the other eight pixels surrounding x_9 shown in Fig. 1(b) are selected to form sub-set $\mathbf{x_{n-1}}$.

Correspondingly, the embedding idea is also partitioned into two parts. The first one is the process of obtaining the detailed classification of $\mathbf{x_{n-1}}$ and then producing the corresponding embedding strategy for $\mathbf{x_{n-1}}$. The other one is the process of using the mean value, denoted by the notation $\mathbf{\bar{x}_{n-1}}$ of the first $(n-1)$ pixels to predict the remaining pixel x_n.

3.1. **Process of producing classification and embedding strategy for each $\mathbf{x_{n-1}}$.** The mean value $\mathbf{\bar{x}_{n-1}}$ of sub-set $\mathbf{x_{n-1}}$ is calculated using the following formula.

$$\mathbf{\bar{x}_{n-1}} = \lfloor \frac{x_1 + x_2 + \cdots + x_{n-1}}{n-1} \rfloor \tag{3}$$

The relation between $\mathbf{\bar{x}_{n-1}}$ and vT_h is used to estimate whether each sub-block has strong inter-pixels correlation or not. If $|\mathbf{\bar{x}_{n-1}} - x_n| \leq vT_h$, then this $\mathbf{x_{n-1}}$ is regarded as smooth one, where vT_h is a predefined threshold which is used for distinguishing which classification sub-set $\mathbf{x_{n-1}}$ belongs to. Otherwise, $\mathbf{x}$ is a complex sub-set.

Alattar's idea based on non-variability of the mean value is applied to $\mathbf{x_{n-1}}$ to create $(n-2)$ difference values. The detailed integer transform $T(\cdot)$ is described as follows.

$$\begin{array}{ll} \mathbf{\bar{x}_{n-1}} & = \lfloor \frac{x_1+x_2+\cdots+x_{n-1}}{n-1} \rfloor \\ d_1 & = x_2 - x_1 \\ d_2 & = x_3 - x_2 \\ \cdots & \\ d_{n-2} & = x_{n-1} - x_{n-2} \end{array} \tag{4}$$

The inverse integer transform $T^{-1}(\cdot)$ is defined as follows.

$$\begin{array}{ll} x_1 & = \mathbf{\bar{x}_{n-1}} - \lfloor \frac{(n-2)\times d_1 + (n-3)\times d_2 + \cdots + 2d_{n-3} + d_{n-2}}{n-1} \rfloor \\ x_2 & = d_1 + x_1 \\ x_3 & = d_2 + x_2 \\ \cdots & \\ x_{n-1} & = d_{n-2} + x_{n-2} \end{array} \tag{5}$$

For a smooth sub-set, if its difference value d_i falls into the range of $[-pT_h, pT_h)$, d_i is expanded twice according to Eq. 6, where $1 \leq i \leq (n-2)$ and $i \in \mathbb{Z}$, pT_h is used for estimating the correlation between two neighboring pixels, e_i represents 2-bit watermark, i.e., $e_i \in \{0, 1, 2, 3\}$.

$$d_i^{'} = 4 \times d_i + e_i \tag{6}$$

If $d_i \geq pT_h$ and $d_i \leq -pT_h - 1$, it will be shifted according to the following formula

$$d_i^{'} = \begin{cases} d_i - 3 \times pT_h & d \leq -pT_h - 1 \\ d_i + 3 \times pT_h, & d \geq pT_h \end{cases} \tag{7}$$

For a complex sub-set, if its difference value d_i falls into the range of $[-pT_h, pT_h)$, d_i is expanded only once based on Eq. 8, where $1 \leq i \leq n-2$ and $i \in \mathbb{Z}$.

$$d_i^{'} = 2 \times d_i + b_i \tag{8}$$

where $b_i \in \{0, 1\}$. If $d_i \geq pT_h$ and $d_i \leq -pT_h - 1$, d_i will be shifted according to the following formula

$$d_i^{'} = \begin{cases} d_i - pT_h & d \leq -pT_h - 1 \\ d_i + pT_h, & d \geq pT_h \end{cases} \tag{9}$$

3.2. **Predicting pixel x_n using the mean value $\bar{\mathbf{x}}_{\mathbf{n-1}}$.** The pixel x_n is predicted using the mean value $\bar{\mathbf{x}}_{\mathbf{n-1}}$ to obtain the prediction-error. Assume d_n to be difference value between x_n and $\bar{\mathbf{x}}_{\mathbf{n-1}}$, the embedding process is described as follows

$$y_n = \begin{cases} \bar{\mathbf{x}}_{\mathbf{n-1}} + 2d_n + b_n & d_n \in [-pT_h, pT_h) \\ x_n - pT_h & d_n \leq -pT_h - 1 \\ x_n + pT_h, & d_n \geq pT_h \end{cases} \tag{10}$$

3.3. **Comparison with Wang *et al.*'s method.** Since $\bar{\mathbf{x}}_{\mathbf{n-1}} = \bar{\mathbf{y}}_{\mathbf{n-1}}$, substitute it and Eq. 8 into Eq. 5, we have

$$\begin{array}{ll} y_1 & = \bar{\mathbf{y}}_{\mathbf{n-1}} - \lfloor \frac{d_{\sum}+b_{\sum}}{n-1} \rfloor \\ y_2 & = 2d_1^{'} + y_1 \\ y_3 & = 2d_2^{'} + y_2 \\ \cdots & \\ y_{n-1} & = 2d_{n-2}^{'} + y_{n-2} \end{array} \tag{11}$$

where $d_{\sum} = 2((n-2) \times d_1 + (n-3) \times d_2 + \cdots + 2d_{n-3} + d_{n-2})$, $b_{\sum} = (n-2) \times b_1 + (n-3) \times b_2 + \cdots + b_{n-2}$. After simplification, $d_{\sum}$ is equal to $2(\Sigma_{i=0}^{n-1} x_i - (n-1) \times x_1)$. On the other hand, since $b_i \in \{0, 1\}$, $1 \leq i \leq n-2$ and $i \in \mathbb{Z}$, then $0 \leq b_{\sum} \leq \frac{(n-1)\times(n-2)}{2}$ and $b_{\sum} \in \mathbb{Z}$. Suppose $x_1 + x_2 + \cdots + x_{n-1}$ equals $(n-1)k_1 + k_2$, where k_1 is actually $\bar{\mathbf{x}}_{\mathbf{n-1}}$, $k_2 \in \{0, 1, \cdots, n-2\}$. Substitute this resulting into Eq. 11, $y_1 = 2x_1 - \bar{\mathbf{x}}_{\mathbf{n-1}} - \lfloor \frac{b_{\sum}+2k_2}{n-1} \rfloor$, $y_2 = 2x_2 - \bar{\mathbf{x}}_{\mathbf{n-1}} - \lfloor \frac{b_{\sum}+2k_2}{n-1} \rfloor + b_1$, $\cdots$, $y_{n-1} = 2x_{n-1} - \bar{\mathbf{x}}_{\mathbf{n-1}} - \lfloor \frac{b_{\sum}+2k_2}{n-1} \rfloor + \sum_{i=1}^{n-2} b_i$, where $0 \leq \lfloor \frac{b_{\sum}+2k_2}{n-1} \rfloor \leq \lfloor \frac{(n-2)(n+3)}{2(n-1)} \rfloor$, and $\lfloor \frac{b_{\sum}+2k_2}{n-1} \rfloor \in \mathbb{Z}$. The formula $\lfloor \frac{(n-2)(n+3)}{2(n-1)} \rfloor$ itself seems too complex, and hence, the notation d_e is created to denote it. As a result, $2x_1 - \bar{\mathbf{x}}_{\mathbf{n-1}} - d_e \leq y_1 \leq 2x_1 - \bar{\mathbf{x}}_{\mathbf{n-1}}, \cdots$.

For instance, comparing Eq. 1 and the results above, we can see that $y_1 = 2x_1 - (a(\mathbf{x}) + LSB(a(\mathbf{x}))) + w_1$ in Wang *et al.*'s method, while in the proposed method, $2x_1 - \bar{\mathbf{x}}_{\mathbf{n-1}} - d_e \leq y_1 \leq 2x_1 - \bar{\mathbf{x}}_{\mathbf{n-1}}$. In Wang *et al.*'s method, since $a(\mathbf{x})$ is the mean value of $\mathbf{x_n}$, y_1 is converted to $2x_1 - \bar{\mathbf{x}}_{\mathbf{n}} - LSB(\mathbf{x_n}) + w_1$. In Alattar's method, as the value of n is increased and all the to-be-embedded-bits are '1', the value of d_e also becomes larger correspondingly, and is getting more and more closer to $\lfloor \frac{n+3}{2} \rfloor$. Therefore, the distortion introduced by Alattar's method becomes more and more higher than that by Wang *et al.*'s method with the value of n increased. To decrease the distortion induced by Alattar's algorithm itself, $n = 4$ is selected in the proposed method.

3.4. **Data Embedding.** To prevent the overflow/underflow, each watermarked pixel is classified into one of three sets: NO_s, $\mathbf{x}_S$ and $\mathbf{x}_C$. Set $\mathbf{x}_S$ contains all sub-blocks with smooth $\mathbf{x_{n-1}}$ whose all $(n-1)$ watermarked values fall in the range of $[0, 255]$, i.e., $0 \leq y_i \leq 255 (1 \leq i \leq n-1$ and $i \in \mathbb{Z})$ and $0 \leq y_n \leq 255$. And meanwhile, set $\mathbf{x}_C$ contains all sub-blocks with complex $\mathbf{x_{n-1}}$ whose all $(n-1)$ watermarked values fall in the range of $[0, 255]$ and $0 \leq y_n \leq 255$. After any modification above (including Eq. 6 to Eq. 10), if overflow/underflow occurs, i.e., $y_i > 255$ or $y_i < 0 (1 \leq i \leq n$ and $i \in \mathbb{Z})$, then this sub-block belongs to set NO_s.

A location map is generated in which the locations of the sub-blocks belongs to NO_s are marked by '1' while the others are marked by '0'. The location map is compressed losslessly by an arithmetic encoder and the resulting bitstream is denoted by $\mathcal{L}$. L_S is the bit length of $\mathcal{L}$.

For $\mathbf{x} \in \mathbf{x}_S$, if any difference value $d_i (i \in \{1, 2, \cdots, n-2\}$ of sub-set $\mathbf{x_{n-1}}$ falls in the range of $[-pT_h, pT_h)$, then it is classified into E_2, otherwise, it belongs to H_2. Similarly, for $\mathbf{x} \in \mathbf{x}_C$, if $d_i \in [-pT_h, pT_h)$, then this d_i is classified into E_1. Otherwise, it is classified into H_1. For x_n of sub-blocks belonging to set $\mathbf{x}_S \bigcup \mathbf{x}_C$, if its corresponding watermarked value y_n obtained from Eq. 10 does not suffer from overflow/underflow, and meanwhile, $d_{n-1} = x_n - \bar{\mathbf{x}}_{\mathbf{n-1}}$ falls inside the interval $[-pT_h, pT_h)$, then 1-bit watermark is embedded into x_n. For the convenience of description, this x_n is grouped into set E_3, otherwise, it belongs to H_3. For sub-set $\mathbf{x}_{\mathbf{n-1}}$ belonging to E_2, each can at most carry 2(n-1) watermark bit, so the maximum hiding capacity is $C_{ap} = 2\|E_2\| + \|E_1\| + \|E_3\| - L_S$ bits, where $\| \cdot \|$ represents the cardinality of a set.

For each sub-block, if it is in NO_s, then $\mathbf{x}$ is kept unaltered, i.e., $\mathbf{y} = \mathbf{x}$. If $\mathbf{x}_{n-1}$ belongs to set E_2, then it is expanded twice according to Eq. (6). If $\mathbf{x}_{n-1}$ belongs to H_2, it will be shifted according to Eq. (7). If $\mathbf{x}$ belongs to set E_1, then it is expanded only once according to Eq. (8). If $\mathbf{x}_{n-1}$ belongs to H_1, it will be shifted according to Eq. (9). If x_n belongs to set $E_3 \cup H_3$, then it is modified according to Eq. (10).

After the first L_S pixels have been processed, the LSBs of y is appended to the payload $\mathcal{P}$ and replaced by the compressed location map $\mathcal{L}$. After all the sub-blocks are processed, a new marked image I_w is obtained.

3.5. **Data Extraction and Image Restoration.** The LSBs of the pixels in I_w are collected into a bitstream $\mathcal{B}$ according to the same order as in embedding. $\mathcal{B}$ is decompressed by an arithmetic decoder to retrieve the location map.

For each watermarked sub-block, if its location is associated with ‘0’ in the location map, then it is ignored. Otherwise, if x_n is retrieved as following.

$$x_n = \begin{cases} \lfloor \frac{y_n - \bar{\mathbf{x}}_{\mathbf{n-1}}}{2} \rfloor + \bar{\mathbf{x}}_{\mathbf{n-1}} & d'_{n-1} \in [-2pT_h, 2pT_h - 1] \\ y_n + pT_h & d'_{n-1} \leq -2pT_h - 1 \\ y_n - pT_h, & d'_{n-1} \geq 2pT_h \end{cases} \tag{12}$$

where $d'_{n-1} = y_n - \mathbf{x}_{\mathbf{n-1}}$. Correspondingly, the embedded watermark is extracted by the following formula: $b_i = y_n - 2\lfloor \frac{y_n - \bar{\mathbf{x}}_{\mathbf{n-1}}}{2} \rfloor$.

After x_n is restored, $d_{n-1} = x_n - \bar{\mathbf{x}}_{\mathbf{n-1}}$ is calculated, if $|d_{n-1}| \leq vTh$, then x_i is retrieved as follows.

$$d_i = \begin{cases} \lfloor \frac{d'_i}{4} \rfloor & d'_i \in [-4pT_h, 4pT_h - 1] \\ d'_i + 3 \times pT_h & d'_i \leq -4pT_h - 1 \\ d'_i - 3 \times pT_h, & d'_i \geq 4pT_h \end{cases} \tag{13}$$

where $d'_i = y_{i+1} - y_i$ and $i \in \{1, 2, \cdots, n-2\}$. Correspondingly, the embedded watermark is extracted by the following formula: $e_i = d'_i - 4\lfloor \frac{d'_i}{4} \rfloor$. Otherwise, x_i is retrieved as follows.

$$d_i = \begin{cases} \lfloor \frac{d'_i}{2} \rfloor & d'_i \in [-2pT_h, 2pT_h - 1] \\ d'_i + pT_h & d \leq -2pT_h - 1 \\ d'_i - pT_h, & d \geq 2pT_h \end{cases} \tag{14}$$

where $d'_i = y_{i+1} - y_i$ and $i \in \{1, 2, \cdots, n-2\}$. Correspondingly, the embedded watermark is extracted by the following formula: $e_i = d'_i - 2\lfloor \frac{d'_i}{2} \rfloor$. Finally, the retrieved difference valuess and the mean value $\bar{\mathbf{x}}_{\mathbf{n-1}}$ are substituted into Eq. 5 to obtain the original pixel value x_i.

4. **Experimental results.** The capacity vs. distortion comparisons among the proposed method, Wang *et al.*’s, Weng’s and Alg. D2 are shown in Figs. 2 and 5. Triple embedding is performed in Weng’s method and Alg. D2 to achieve the embedding bit-rate of over 1.0 bpp.

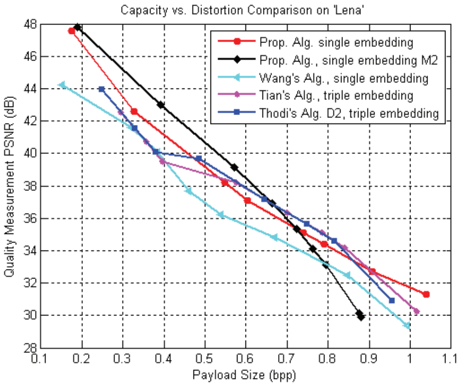


FIGURE 2. Capacity vs. Distortion Comparison on 'Lena'

Since Wang *et al.*'s method with 4 × 4 sized block performs the best, the location map only is $\frac{1}{16}$ size, relative to any original grayscale image. And therefore, small embedding capacity still can be obtained even when the threshold is set to a small integer value. And meanwhile, Wang *et al.*'s method only modified the embeddable sub-block while making almost no modifications to the other sub-blocks. Therefore, the embedding-induced distortion can be kept at a low level, even if the embedding bit-rate is low. However, with embedding bit-rate increased, the distortion is increasingly severe. In the proposed method, for any smooth block, if its difference value between two neighboring pixels falls into the range of $[-pT_h, pT_h)$, then it will be expanded twice. For the other smooth blocks, their pixel values must be shifted so as to ensure reversibility. In a word, the reversibility is achieved with additional costs of distortion. Hence, at the low embedding rates, Wang *et al.*'s method provides better performance than the proposed method. However, the number of smooth blocks used in shifting is decreased as the embedding rate is increased. The proposed method performs well, and significantly outperforms Wang *et al.*'s method for larger embedding rates. For 'Lena'image, it also can be seen from Fig. 2, relative to Wang *et al.*'s method and the proposed method, both higher embedding rates and larger PSNR values are obtained by Weng's and Alg. D2 exceeding 0.7 bpp. The proposed method exceeds all methods at about 0.9 bpp.

Baboon' is a typical image with large areas of complex texture, so the obtained bit-rate is slightly lower at the same PSNR. Fig. 3 show that the proposed method achieves higher embedding capacity with lower embedding distortion than the others. The proposed method have the same performance with the other methods.

For 'Barbara' images, it can be seen from Fig. 4 that the proposed method have better performance than Weng's method and Alg. D2 at almost all embedding rates. When the embedding rate is over 0.55 bpp, the proposed method, Weng 's method and Alg. D2 all exceed Wang *et al.*'s method in both embedding rates and PSNR values. As a result, the proposed method can achieve larger embedding rates while keeping higher PSNR in the field of reversible watermarking based on integer transform. There is similar situation for 'Goldhill' image (shown in Fig. 5).

5. **Conclusions.** A new reversible watermark scheme capable of mostly carrying $2n - 3$ bits into one n-sized image block in a single embedding process have been presented in this paper. With a variant relation between one pixel and the mean value of the other $(n - 1)$ pixels, all the sub-blocks are divided into two classes: smooth block and complex block. More bits are embedded into smooth blocks while fewer bits into complex blocks.

Through a series of experiments, the proposed method has better performance than the other methods at larger embedding rates.

Acknowledgment. This work was supported in part by National NSF of China (No. 61201393, No. 61001179).

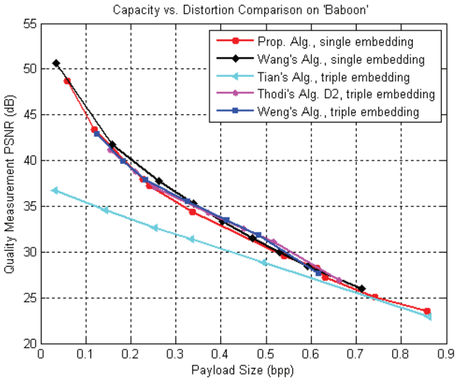


Figure 3. Capacity vs. Distortion Comparison on 'Baboon'

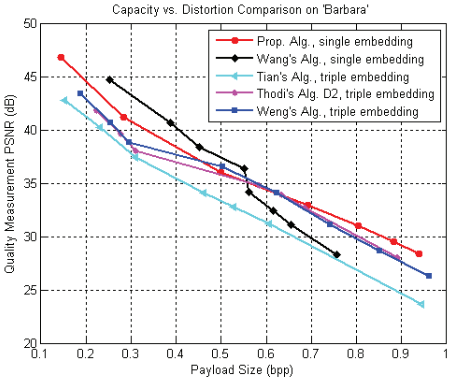


Figure 4. Capacity vs. Distortion Comparison on 'Barbara'

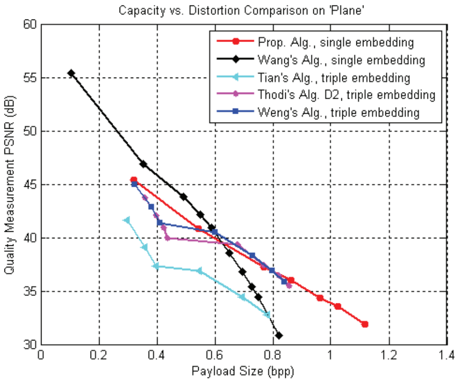


Figure 5. Capacity vs. Distortion Comparison on 'Plane'

REFERENCES

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

[2] A. M. Alattar, Reversible watermark using the difference expansion of a generalized integer transform, *IEEE Trans. Image Processing*, vol. 13, no. 8, pp. 1147-1156, 2004.

[3] D. Coltuc, and J. M. Chassery, Very fast watermarking by reversible contrast mapping, *Journal of IEEE Signal Processing Letters*, vol. 14, no. 4, pp. 255-258, 2007.

[4] D. M. Thodi, and J. J. Rodriguez, Expansion embedding techniques for reversible watermarking *IEEE Trans. Image Processing*, vol. 16, no. 3, pp. 721-730, 2007.

[5] S. W. Weng, Y. Zhao, J. S. Pan, and R. R. Ni, Reversible watermarking based on invariability and adjustment on pixel pairs, *Journal of IEEE Signal Processing Letters*, vol. 15, pp. 721-724, 2008.

[6] S. W. Weng, Y. Zhao, R. R. Ni, and J. S. Pan, Parity-invariability-based reversible watermarking, *Journal of Electronics letters*, vol. 45, no. 20, pp. 1022-1023, 2009.

[7] X. Wang, X. L. Li, B. Yang, and Z. M. Guo, Efficient generalized integer transform for reversible watermarking, *Journal of IEEE Signal Processing Letters*, vol. 17, no. 6, pp. 567-570, 2010.