Computers, Materials & Continua CMC, vol.62, no.1, pp.157-177, 2020

A General Framework of Reversible Data Hiding with Controlled Contrast Enhancement

Shaowei Weng[1, 2, *], Yiyun Liu[1], Yunqing Shi[3], Bo Ou[4], Chunyu Zhang[5] and Cuiping Wang[6]

Abstract: This paper proposes a two-step general framework for reversible data hiding (RDH) schemes with controllable contrast enhancement. The first step aims at preserving visual perception as much as possible on the basis of achieving high embedding capacity (EC), while the second step is used for increasing image contrast. In the second step, some peak-pairs are utilized so that the histogram of pixel values is modified to perform histogram equalization (HE), which would lead to the image contrast enhancement. However, for HE, the utilization of some peak-pairs easily leads to over-enhanced image contrast when a large number of bits are embedded. Therefore, in our proposed framework, contrast over-enhancement is avoided by controlling the degree of contrast enhancement. Since the second step can only provide a small amount of data due to controlled contrast enhancement, the first one helps to achieve a large amount of data without degrading visual quality. Any RDH method which can achieve high EC while preserve good visual quality, can be selected for the first step. In fact, Gao et al.'s method is a special case of our proposed framework. In addition, two simple and commonly-used RDH methods are also introduced to further demonstrate the generalization of our framework.

Keywords: Reversible data hiding, controlled contrast enhancement, general framework, PEE-based RDH method, IT-based RDH method.

1 Introduction

As an important research area of information security, data hiding has been studied for years as an efficient protection for multimedia carriers. Most of data hiding techniques focus on correct extraction of hidden data bits, and yet the data embedding process usually introduces permanent distortion for original carriers. However, some special applications, such as in military, legal and medical usages, permit no permanent

[1] School of Information Engineering, Guangdong University of Technology, Guangzhou, 510006, China.

[2] Guangdong Key Laboratory of Intelligent Information Processing and Shenzhen Key Laboratory of Media Security, Shenzhen University, 518060, China.

[3] Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07103 USA.

[4] College of Computer Science and Electronic Engineering, Hunan University, Changsha, 410082, China.

[5] School of Information Engineering, Xizang Minzu University, Xianyang, 712082, China

[6] School of Electronic Information, Qingdao University, Qingdao, 266071, China.

[*] Corresponding Author: Shaowei Weng. Email: wswweiwei@126.com.

distortion to original carriers. To this end, data hiding with reversibility, i.e., RDH, is proposed to satisfy the demand of these applications. Reversibility aims at lossless recovery of original carrier and correct extraction of hidden data.

A large amount of research has been conducted in the past few decades to develop RDH, and thus, some influential research works in the spatial domain have been proposed, including RDH based on lossless compression [Fridrich, Goljan and Du (2002); Celik, Sharma, Tekalp et al. (2005)], RDH using difference expansion (DE) [Tian (2003)], RDH using histogram shifting (HS) [Ni, Shi, Ansari et al. (2006); Xuan, Yang, Zhen et al. (2004); Li, Li, Yang et al. (2013)], RDH using prediction-error expansion (PEE) [Sachnev, Kim, Nam et al. (2009); Li, Yang and Zeng (2011); Hong, Chen and Wu (2013)], and RDH using integer-to-integer transform [Tian (2003); Alattar (2004); Wang, Li, Yang et al. (2010); Wang, Li and Yang (2010); Weng and Pan (2016)]. Besides these RDH methods in the spatial domain, the RDH ones in the encrypted domain have also been developed over years [Xiong and Qing (2018); Chen, Yin, He et al. (2018)], since the first RDH one was proposed by Zhang [Zhang (2011)]. In those traditional RDH methods, PSNR (Power Signal-to-Noise Ratio) is the most commonly-used quality measure for evaluating the visual quality of watermarked images. It is well known that the embedding methods achieving high PSNR values may not lead to high visual quality. To this end, they have presented a RDH scheme with contrast enhancement to achieve high visual quality [Wu, Dugelay and Shi (2015)]. Their advantage is to utilize HE, a commonly-used contrast enhancement, to embed data into two peaks (i.e., a peak-pair) using HS. In this way, multiple peak-pairs are selected and modified to achieve both high EC and contrast enhancement. However, when more and more payload is required, Wu *et al.*'s method may need to select over 50 pairs for data embedding, which may result in over-enhanced contrast. In order to solve the above problem, Gao et al. proposed a RDH scheme with controlled contrast enhancement and Haar integer wavelet transform (IWT) [Gao and Shi (2015)]. Unlike Wu et al.'s method that splits a number of peak-pairs to achieve high EC and HE while ignores the degree of contrast enhancement, Gao et al.'s method controls the number of peak-pairs to avoid over-enhanced contrast. The control on the number of peak-pairs will lead to the decrease of EC. To this end, IWT is used to achieve high EC. More importantly, IWT can produce very small contrast change. In this way, Gao et al.'s method not only increases EC, but also maintains satisfactory visual quality. In fact, besides IWT, some other RDH methods can also be chosen for achieving higher EC under the same visual quality.

Our proposed framework is based on the fact that most PEE-based (or integer-transform-based (IT-based)) RDH methods make small modifications to image histogram, and especially, when the required EC is not very high, the image histogram is almost not changed before and after data embedding. Inspired by this fact, we utilize respectively two previous RDH methods which can provide enough EC (e.g., about 0.7 bpp) and maintain high image quality, to illustrate the generalization of our proposed framework. Since the data embedding process in the first step does not introduce too much changes for image histogram, the watermarked image can be used again in the second step to increase image contrast by HE. By incorporating these two steps, both EC and image contrast are improved. Specifically, in the first step, two selected RDH methods are used for providing a large part

of the payload while maintaining high visual quality. One is a simple and representative work among existing PEE-based RDH methods, i.e., the RDH method based on the Gap (gradient-adjusted prediction) [Wu and Memon (1997)] and an embedding-position-selection strategy. The other is our recent work [Weng and Pan (2016)] incorporating block selection into the integer transform proposed by Alattar [Alattar (2004)]. The second step is used for achieving contrast enhancement and embedding the rest payload. It is expected that more efficient RDH methods can be devised according to the proposed framework by carefully designing RDH methods.

2 Proposed RDH scheme

The main purpose of our proposed method is to construct a general framework of RDH with the contrast enhancement. As shown in Fig. 1, the generalization of our proposed method lies in the fact that any high-performance RDH method can be applied in our framework. Furthermore, our proposed method is still a RDH method. Usually, the essence of any RDH method is to hide data losslessly into a host image for some special applications, such as medical image processing. This is also the reason that both data hiding and contrast enhancement are needed in our framework. There are two main steps in our proposed framework. In the first step, two RDH methods (i.e., the PEE-based and IT-based methods) which are described respectively in Sections 2.1 and 2.2, are selected for achieving the required EC while maintain high visual quality. By these two methods, we want to illustrate that any RDH method which can provide enough EC (e.g., 0.7 bpp) while maintain high visual quality, can be used for our framework. HE, given in Section 2.3, is used in the second step to enhance contrast.

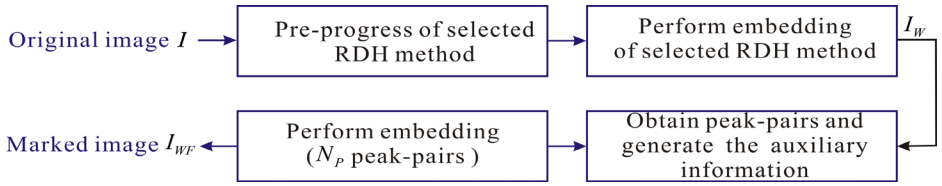


Figure 1: The diagram of the proposed scheme

2.1 IT-based RDH method

The RDH method utilized in our recent work can evaluate accurately the local complexity of blocks by incorporating the mean values of blocks into the evaluation of complexity [Weng and Pan (2016)]. Since the mean values of blocks are used for evaluating the local complexity, they must be kept unaltered so as to ensure reversibility. This is the reason that the integer transform proposed by Alattar [Alattar (2004)] is utilized in our proposed work.

A test image I of size $W_R \times W_C$ is split into disjoint n-sized image blocks: $B_1, \cdots, B_N$ in raster scan order (from top to bottom and left to right), where $n = r \times c$, r and c are the height and width of blocks, respectively, and N is the total number of blocks, i.e.,

$N = \left\lfloor \frac{W_R}{r} \right\rfloor \times \left\lfloor \frac{W_C}{c} \right\rfloor$. For the ease of description, we use the notation B to denote one of N blocks. All the pixels in the block B are arranged into a vector $\mathbf{x} = \{x_1, \cdots, x_n\}$ according to a predefined order.

The transform proposed by Alattar is defined as follows

$$\begin{aligned} \bar{\mathbf{x}} &= \left\lfloor \frac{x_1 + x_2 + \cdots + x_n}{n} \right\rfloor, \\ d_1 &= x_2 - x_1, \\ d_2 &= x_3 - x_2, \\ &\cdots \\ d_{n-1} &= x_n - x_{n-1}, \end{aligned} \tag{1}$$

where $\bar{\mathbf{x}} = \left\lfloor \frac{x_1 + x_2 + \cdots + x_n}{n} \right\rfloor$, d_k is the difference value between two adjacent pixels x_{k+1} and x_k, $k \in \{1, \cdots, n-1\}$. The inverse integer transform of Eq. (1) is defined as

$$\begin{aligned} x_1 &= \bar{\mathbf{x}} - \left\lfloor \frac{(n-1) \times d_1 + (n-2) \times d_2 + \cdots + d_{n-1}}{n} \right\rfloor, \\ x_2 &= x_1 + d_1, \\ x_3 &= x_2 + d_2, \\ &\cdots \\ x_n &= x_{n-1} + d_{n-1}. \end{aligned} \tag{2}$$

Each of $n-1$ difference values is modified according to Eq. (3). Specifically, if $d_k \in [-pT_h, pT_h)$, then d_k is termed embeddable difference value, namely it is embedded with 1-bit watermark so that its marked value d'_k is generated by Eq. (3); if $d_k \in (-\infty, -pT_h) \bigcup [pT_h, \infty)$, d_k is shifted by pT_h to obtain d'_k according to Eq. (3).

$$d'_k = \begin{cases} 2 \times d_k + b, & -pT_h \le d_k < pT_h, \\ d_k - pT_h, & d_k \le -pT_h - 1, \\ d_k + pT_h, & d_k \ge pT_h, \end{cases} \tag{3}$$

where pT_h is a predefined threshold which is used to classify difference values into two groups: embeddable ($\left[-pT_h, pT_h\right)$) and shifted ($(-\infty, -pT_h) \bigcup [pT_h, \infty)$), and b stands for 1-bit hidden data, i.e., $b \in \{0,1\}$.

Afterwards, d_k in Eq. (2) is substituted by d'_k to obtain the marked pixel list $\mathbf{y} = \{y_1, \cdots, y_n\}$ (refer to Eq. (4)).

$$y_1 = \bar{\mathbf{x}} - \frac{(n-1)\times d'_1 + (n-2)\times d'_2 + \cdots + d'_{n-1}}{n}, \tag{4}$$
$$y_2 = y_1 + d',$$
$$y_3 = y_2 + d',$$
$$\cdots$$
$$y_n = y_{n-1} + d'_{n-1}.$$

2.1.1 Smoothness classification

In order to improve estimation accuracy of the local complexity, the average value $\bar{\mathbf{x}}$ of a block B together with $r+c+1$ pixels in the neighborhood of B (i.e., $x_{1,c+1},\cdots,x_{r,c+1},x_{r+1,c+1},x_{r+1,1},\cdots,x_{r+1,c}$) constitute a set I_{ENP} and are used for evaluating the local complexity. The local complexity Δ is calculated as the variance of set using the equation

$$\Delta = \sqrt{\frac{\sum_{k=1}^{r+1}(x_{k,c+1}-u_{ENP})^2 + (\bar{\mathbf{x}}-u_{ENP})^2 + \sum_{k=1}^{c}(x_{r+1,k}-u_{ENP})^2}{r+c+2}}, \tag{5}$$

where μ_{ENP} is the average value of set I_{ENP}, and vT_h is a predefined threshold used for separating all the blocks into two groups: smooth ($\Delta < vT_h$) and rough ($\Delta \geq vT_h$).

2.1.2 Additional information

If one pixel whose marked value exceeds the range of $[0,255]$, it is deemed as one overflow/underflow pixel, and thus needs to be excluded from data embedding. Similarly, the blocks containing overflow/underflow pixels also need to be removed from data embedding. O_{s2} is used to indicate a set, which is composed of the blocks with $\Delta \geq vT_h$. The blocks with $\Delta < vT_h$ are classified into two sets E_s and O_{s1}. Specifically, E_s contains the blocks without overflow/underflow pixels, while O_{s1} includes the blocks with overflow/underflow pixels. Since the blocks in O_{s2} can be easily distinguished from the ones in $E_s \cup O_{s1}$ by comparing Δ with the threshold vT_h, their locations are not needed to be recorded in a location map. Therefore, a location map is created to only differentiate the blocks in E_s from the ones in O_{s1}. Afterwards, the location map is compressed losslessly into a bitstream $\mathcal{L}$ by an arithmetic encoder. L_{SC} is the length of $\mathcal{L}$.

Besides the bitstream $\mathcal{L}$, some other additional information including vT_h (8 bits), pT_h (8 bits), and block size parameters r (2 bits), c (2 bits) is helpful for blind data extraction and image restoration. Therefore, all these additional information along with the payload is needed to be inserted into the host image I. The size of the additional information is supposed to be L_Σ bits. The maximum embeddable payload equals the available payload minus L_Σ, i.e., $\mathbf{P_M} = N_{<pT_h}$ - L_Σ, where $N_{<pT_h}$ stands for the number of embeddable

differences whose values belong to the range of $[-pT_h, pT_h)$.

2.1.3 Data embedding

For better illustration, $\mathcal{P}_C$ is used to represent the to-be-embedded payload. The procedure of embedding the payload $\mathcal{P}_C$ into the host image I is described below.

Step 1 Watermark embedding

if $\mathbf{x} \in E_s$

$\mathbf{y}$ is obtained using Eq. (4);

elseif $\mathbf{x} \in O_{s1} \cup O_{s2}$

$\mathbf{y} = \mathbf{x}$;

end

Step 2 Generating the marked image I_W

The data embedding process is implemented in a block-wise manner. N_1 blocks in the host image I are first modified in the raster scan order, where $N_1 \ll N$. Suppose the used payload is $\mathcal{P}_{C1}$, where $\mathcal{P}_{C1} \ll \mathcal{P}_C$. Next, the resulted image is scanned in a pixel-wise manner according to raster scan order, the LSBs of the first L_Σ scanned pixels are collected into a bitstream C_1, and simultaneously each empty LSB is replaced by one of the L_Σ additional bits. Then, except for N_1 already-modified blocks, the remaining ones are modified according to Step 1 until the remaining payload $\mathcal{P}_C - \mathcal{P}_{C1}$ and the bitstream C_1 are embedded into the host image I. Finally, a watermarked image I_W is generated.

2.1.4 Image restoration and data extraction

The pixels in the marked image I_W is processed one by one in raster scan order, and their LSBs are collected to form a bitstream $\mathcal{B}$. An arithmetic decoder is used to decompress $\mathcal{B}$ so that the location map is recovered. The location map is re-compressed by an arithmetic encoder into a bitstream whose length is L_{SC}. Once L_{SC} is known, pT_h, vT_h, r and c are extracted from the bitstream $\mathcal{B}$ according to their own lengths, respectively. The marked image I_W is separated into $r \times c$ -sized non-overlapped image blocks: $B_1^w, \cdots, B_N^w$ according to raster scan order. In order to correctly extract data bits and retrieve host images, the extraction order of blocks is contrary to the embedding one of blocks, i.e., $B_N^w, \cdots, B_1^w$.

For the simplicity of description, B^w is used to indicate one of N marked blocks. If the neighborhood surrounding B^w is composed of $(r+c+1)$ neighbors, B^w is used for data extraction. It should be mentioned that the $(r+c+1)$ neighbors are either original pixels or already-extracted pixels. For each of the blocks used for data embedding, its Δ is

calculated via Eq. (5). For a block with $\Delta \geq vT_h$, it remains unaltered, i.e., $\mathbf{x} = \mathbf{y}$. For a block with $\Delta < vT_h$, if its corresponding bit in the location map is 1, then it is skipped; if its corresponding bit is 0, then the hidden bit b is extracted according to the following formula $b = \mathrm{mod}(d'_k, 2)$: where $d'_k \in [-2pT_h, 2pT_h - 1]$, and the original difference value is calculated using Eq. (6).

$$d_k = \begin{cases} \dfrac{d'_k}{2}, & -2pT_h \leq d'_k \leq 2pT_h - 1, \\ d'_k + pT_h, & if\ d'_k \geq 2pT_h, \\ d'_k - pT_h, & ifd'_k \leq -2pT_h - 1. \end{cases} \tag{6}$$

2.2 PEE-based RDH method

Considering that the method in Section 2.1 is executed in a block-by-block manner, a new RDH method in a pixel-wise manner is selected so as to better illustrate the generalization of our proposed framework. This adopted method uses the Gap predictor to generate prediction. Besides, the prediction-errors located in smooth regions are priorly chosen for data embedding. For a pixel x, its complexity denoted as Δ is calculated using its context C_{text} containing 7 right and bottom neighbors as shown in Fig. 2. Specifically, Δ is defined as the variance of the pixels in C_{text}.

$$\Delta = \sqrt{\frac{\sum_{i \in \{1,\cdots,7\}} (c_i - \mu)^2}{7}}, \tag{7}$$

where μ is the mean value of set C_{text}. A threshold vT_h is used for splitting all pixels into two groups: smooth ($\Delta < vT$) and rough ($\Delta \geq vT_h$).

	$j-1$	j	$j+1$	$j+2$
i		x	c_1	c_2
$i+1$	c_3	c_4	c_5	
$i+2$	c_6	c_7		

Figure 2: x is the current pixel, $C_{text} = \{c_i : i \in \{1,2,\cdots,6,7\}\}$ contains 7 right and down neighbors of x

Referring to Fig. 2, C_{text} is reused to predict x using the Gap predictor, and the prediction value $\hat{x}$ is defined using the equation

$$\hat{x}=\begin{cases} c_1, & d_d>80, \\ \dfrac{c_1+x^*}{2}, & d_d\in(32,80], \\ \dfrac{c_1+3x^*}{4}, & d_d\in(8,32], \\ x^*, & d_d\in[-8,8], \\ \dfrac{c_3+3x^*}{4}, & d_d\in[-32,-8), \\ \dfrac{c_3+x^*}{2}, & d_d\in[-80,-32), \\ c_3, & d_d<-80, \end{cases} \tag{8}$$

where $d_d=d_v-d_h$, $x^*=\dfrac{(c_1+c_3)}{2}+\dfrac{(c_2-c_4)}{4}$, and the vertical and horizontal gradients of x are estimated as $d_v=|c_1-c_4|+|c_2-c_6|+|c_3-c_7|$, $d_h=|c_1-c_5|+|c_2-c_3|+|c_3-c_4|$, respectively.

The corresponding prediction-error is denoted as $e=x-\hat{x}$. For each prediction-error e, it is expanded or shifted as

$$e'=\begin{cases} 2\times e+b, & -pT_h\le e<pT_h, \\ e+pT_h, & e\ge pT_h, \\ e-pT_h, & e<-pT_h, \end{cases} \tag{9}$$

where e' is the modified prediction-error, pT_h is an integer-valued capacity-control parameter, and $b\in\{0,1\}$ is 1 data bit to be embedded. After data embedding, the modified pixel is $x'=\hat{x}+e'$.

For $x_{i,j}(i\in\{R-1,R\},j\in\{1,C-1,C\})$, it has not 7 right and bottom neighbors, and therefore, Gap predictor cannot be used for these pixels to generate prediction. Based on this reason, only $(i\in\{1,\cdots,R-2\},j\in\{2,\cdots,C-2\})$, can be used for data embedding. The detailed data embedding procedure is described in Algorithm 1. Correspondingly, the embedded bit b is extracted and the original x is recovered according to Algorithm 2. Like many other RDH methods, a location map LM with size $(R-2)\times(C-3)$ is needed to record the locations of overflow and underflow pixels. Then, if overflow/underflow occurs, the corresponding value in LM is marked as 1 and otherwise as 0.

Algorithm 1. Embedding.

Input: The subimage excluding boundary pixels: $J=\{x_{i,j}:1\le i\le R,1\le j\le C\}$, the predicted value: $\hat{x}_{i,j}(i\in\{1,\cdots,R-2\},j\in\{2,\cdots,C-2\})$, the prediction-error: $e_{i,j}(i\in\{1,\cdots,R-2\},j\in\{2,\cdots,C-2\})$, the complexity of $x_{i,j}$: $\Delta_{i,j}(i\in\{1,\cdots,R-2\},j\in\{2,\cdots,C-2\})$, the location map: LM, the to-be-embedded bit:

b, two thresholds: vT_h and pT_h.

Output: The marked image: $I_W = \{x'_{i,j} : 1 \le i \le R, 1 \le j \le C\}$.

for $i = 1 : R-2$

  for $j = 2 : C-2$

    if $\Delta_{i,j} < vT_h$ && $LM_{i,j} == 0$

      if $e_{i,j} < pT_h$ && $e_{i,j} \ge -pT_h$

        $e'_{i,j} = 2 \times e_{i,j} + b$

      elseif $e_{i,j} \ge pT_h$

        $e'_{i,j} = e_{i,j} + pT_h$

      elseif $e_{i,j} < -pT_h$

        $e'_{i,j} = e_{i,j} - pT_h$

      end

      $x'_{i,j} = \hat{x}_{i,j} + e'_{i,j}$

    end

  end

end

Algorithm 2. Extraction.

Input: The marked image: $I_W = \{x'_{i,j} : 1 \le i \le R, 1 \le j \le C\}$, the predicted value: $\hat{x}_{i,j} (i \in \{1, \cdots, R-2\}, j \in \{2, \cdots, C-2\})$, the marked prediction-error: $e'_{i,j} (i \in \{1, \cdots, R-2\}, j \in \{2, \cdots, C-2\})$, the complexity of $x'_{i,j}$: $\Delta_{i,j} (i \in \{1, \cdots, R-2\}, j \in \{2, \cdots, C-2\})$, the location map: LM, two thresholds: vT_h and pT_h.

Output: The original image: I, the extracted bit: b.

for $i = 1 : R-2$

  for $j = 2 : C-2$

    if $\Delta_{i,j} < vT_h$ && $LM_{i,j} == 0$

      if $e'_{i,j} \le 2pT_h - 1$ &&$e'_{i,j} \ge -2pT_h$

        $e_{i,j} = \lfloor e'_{i,j} / 2 \rfloor$

        $b = e'_{i,j} - 2 \times \lfloor e'_{i,j} / 2 \rfloor$

      elseif $e'_{i,j} \ge 2pT_h$

$e_{i,j} = e'_{i,j} - pT_h$

elseif $e'_{i,j} < -2pT_h$

$e_{i,j} = e'_{i,j} + pT_h$

end

$x_{i,j} = \hat{x}_{i,j} + e_{i,j}$

end

end

end

2.3 Histogram equalization (HE)

HE is a frequently-used method for contrast enhancement. HE is achieved by carrying out repeatedly the data embedding process, and in a single process, splitting each of two peaks (i.e., one peak-pair) into two neighboring bins with similar heights. In a single data embedding process, P_L (Left) and P_R (Right) are two peak points (two highest frequencies of occurrence) in the image histogram, respectively. It should be mentioned that if there exist more than two peak points with the same occurrence frequency, then the two peaks introducing the minimum distortion are selected as P_L , P_R . The data embedding is implemented by

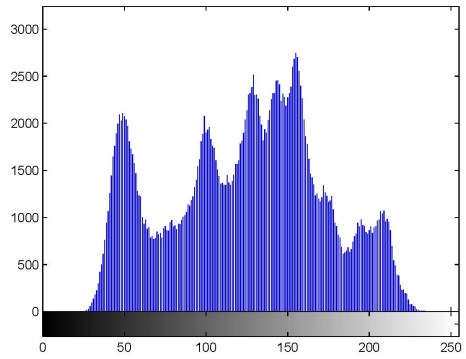
$$x' = \begin{cases} x-1 & \text{if } x < P_L, \\ x-b & \text{if } x = P_L, \\ x & \text{if } P_L < x < P_R, \\ x+b & \text{if } x = P_R, \\ x+1 & \text{if } x > P_R. \end{cases} \tag{10}$$

It is known that it is very difficult for a single embedding process to provide the efficient EC. In order to further increase EC, two peak points in the already-modified histogram continue to be selected for carrying data until the satisfactory contrast enhancement is achieved. In order to effectively avoid overflow/underflow problems, a location map is generated to record the locations of overflow/underflow pixels whose values exceed the range of $[0,255]$. This map needs to be losslessly compressed into a bistream by arithmetic encoding due to its large size. Besides the obtained bitstream, some other auxiliary information including the number of peak pairs (denoted by N_p), the size of compressed location map, and all the peak pairs, should also be embedded into the modified image I_W in the first step for blind decoding. Finally, all auxiliary information along with the required payload is embedded into I_W to obtain the final marked image I_{WF} .

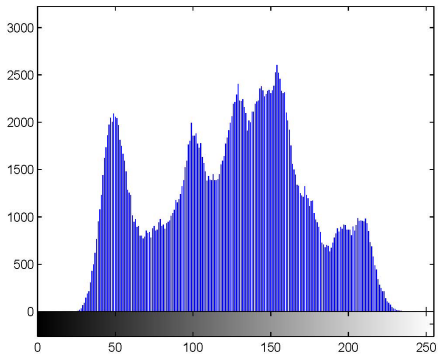
4 Experimental results

The proposed RDH scheme is implemented in MATLAB environment. The Lena, Baboon, Barbara, Baboon, Airplane, Goldhill and Boat images with size 512×512 are provided by the authors of paper [Wu, Dugelay and Shi (2015)]. For ease of description, Wu and Gao are short for Wu et al. and Gao et al., respectively. The image contrast enhancement is evaluated by the relative contrast error (RCE) [Gao and Wang (2013)], and the source codes of calculating RCE and Wu's method are provided by the authors of paper [Wu, Dugelay and Shi (2015)].

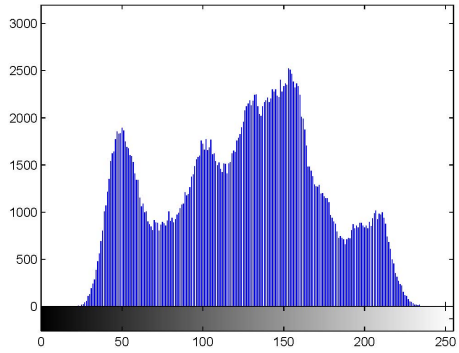
Fig. 3 illustrates that the PEE-based RDH method is used for achieving different EC by setting pT_h and vT_h. From Fig. 3, one can observe that when the EC is not high (e.g., 46637 bits), the modified image histogram is very close to its original one. In the experiments, we simply set $pT_h = 2vT_h$ instead of considering the optimal combination of vT_h and pT_h. For instance, when $pT_h = 2vT_h = 4$, the obtained PSNR is 47.653 (dB) at an EC of 46637 bits (refer to Fig. 3(b)). In order to obtain higher EC, we increase the value of $pT_h = 2vT_h$ from 4 to 6 so as to select more prediction-errors to be used for data embedding. Thus, EC is increased from 46,637 bits to 103,681 bits. As shown in Fig. 3(c), although the image histogram has been changed slightly due to the increase of EC, the outline of the modified image histogram is still very similar to that of the original one. More importantly, the heights of bins in the image histogram are changed slightly when a large amount of data is obtained. This implies that the data embedding process leads to the change of the prediction-error histogram instead of the image histogram. In order to further increase the EC, we set $pT_h = 2vT_h = 9$. Experimental results also show that the obtained capacity is 155,388 bits while higher visual quality is obtained, i.e., PSNR=39.446 dB. From Figs. 3(b) to 3(d), it can be observed that RCE is very small.



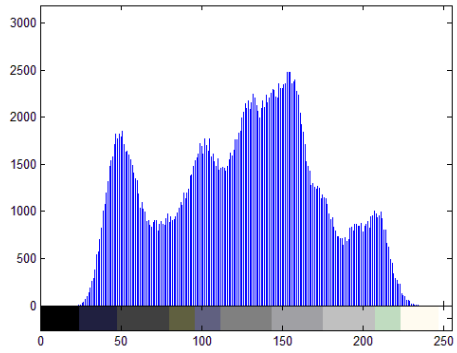
(a) Image histogram before data embedding



(b) Image histogram with PSNR=47.653 (dB), EC=46,637 bits (0.17809 bpp) and RCE=0.50053



(c) Image histogram with PSNR=42.847 (dB), EC=103,681 bits (0.39551 bpp) and RCE=0.50081

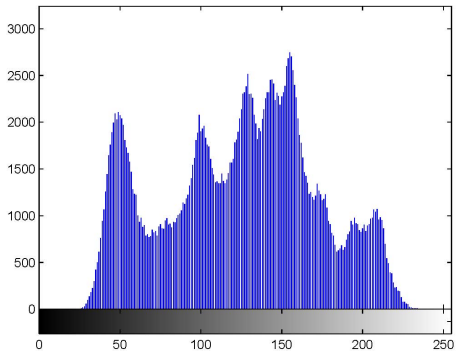


(d) Image histogram with PSNR=39.446 (dB), EC=155,388 bits (0.5429 bpp) and RCE=0.50091

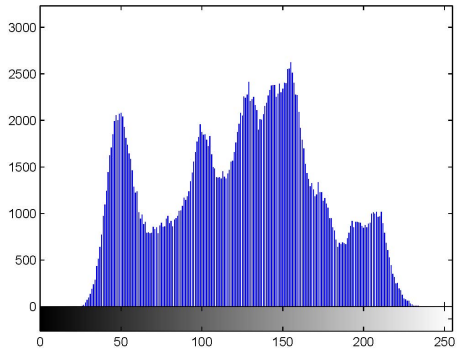
Figure 3: Comparisons of image histogram at different EC for Lena using the PEE-based RDH method

Fig. 4 shows that the embedding process makes the modifications to the image histogram when the IT-based RDH method is selected in the first step. Similarly, IT-based RDH can also achieve high EC while maintain good visual quality, e.g., when the EC is 148,875 bits (0.56791 bpp), PSNR is 39.446 dB. Therefore, the modified image in the first step can be used again in the second step to increase the image contrast.

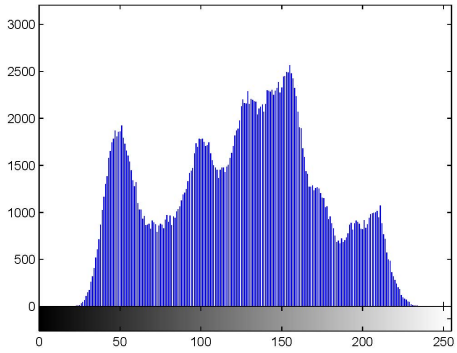
From Figs. 3 and 4, one can conclude that any RDH method which achieves data embedding by modifying prediction-error histogram (or difference histogram) do not make large modifications to the image histogram. In this way, high EC can be achieved while the good visual quality can be maintained.



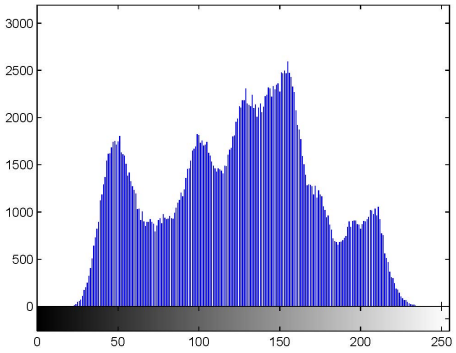
(a) Image histogram before data embedding



(b) Image histogram with PSNR=49.270 (dB), EC=41,286 bits (0.15749 bpp) and RCE=0.50009



(c) Image histogram with PSNR=42.763 (dB), EC=111,114 bits (0.42387 bpp) and RCE=0.50041



(d) Image histogram with PSNR=39.446 (dB), EC=148,875 bits (0.56791 bpp) and RCE=0.50089

Figure 4: Comparisons of image histogram at different EC for Lena using the IT-based RDH method

In Fig. 5, the IT-based method and HE are utilized to achieve high EC and contrast enhancement at the same time. Referring to Fig. 5(b), we set simply $2vT_h = pT_h = 9$ in the first step to achieve an EC of 12,0577 bits with PSNR=42.010 dB. Afterwards, in the second step, 10 peak-pairs are utilized to achieve contrast enhancement and an EC of 43,194 bits at the same time. After two steps, the final EC is 163,771 bits with PSNR=28.642 dB. In order to further improve EC, we set $2vT_h = pT_h = 14$ in the first step and $N_p = 15$ in the second step (refer to Fig. 5(c)). In the first step, 148,775 bits are obtained when PSNR is 39.446 dB. In the following second step, we obtain an EC of 67240 bits. Compared with Fig. 5(a), the image contrast in Figs. 5(b) and 5(c) is obviously enhanced. It also can be observed that the contrast enhancement in Fig. 5(c) is larger than that in Fig. 5(b). In the experimental process, if we simply set $pT_h = 2vT_h$, and select N_p separately, the obtained embedding performance may not be optimal. Therefore, we conclude that there is still a lot of room for improvement of the embedding performance. For example, the optimal combination of vT_h, pT_h and N_p is exhaustively searched so that the visual quality are preserved as much as possible and simultaneously the EC is maximized.

Figure 5: Comparisons of visual perception and embedding capacity of marked images for Lena, Barbara and Goldhill when the IT-based RDH method is used in the first step. (a) Lena. (b) 10 pairs: 28.3417 dB, 189,410 bits, RCE=0.5298. (c) 15 pairs: 25.229 dB, 228,930 bits, RCE=0.5460. (d) Barbara. (e) 10 pairs: 29.606 dB, 124,365 bits, RCE=0.53067. (f) 16 pairs: 25.792 dB, 167,826 bits, RCE=0.54807. (g) Goldhill. (h) 10 pairs: 30.397 dB, 117,591 bits, RCE=0.52507. (i) 17 pairs: 25.412 dB, 178,137 bits, RCE=0.54728

As shown in Fig. 6, the PEE-based method is utilized to achieve high EC and contrast enhancement. In Fig. 6(b), in the first step, we select simply $2vT_h = pT_h = 7$ to achieve an EC of 125,630 bits with PSNR=41.547 dB. Afterwards, in the second step, 10 peak-pairs are utilized to achieve contrast enhancement and the increase of EC (i.e., 42,124 bits). Referring to Fig. 6(c), when $2vT_h = pT_h$ is set to 9, the obtained EC is 155,388 bits and PSNR is 39.889 dB in the first step. In the second step, 16 pairs are used to obtain an EC of 70,271 bits. From Figs. 6(b) and 6(c), the contrast is obviously enhanced.

(a) (b) (c)

(d) (e) (f)

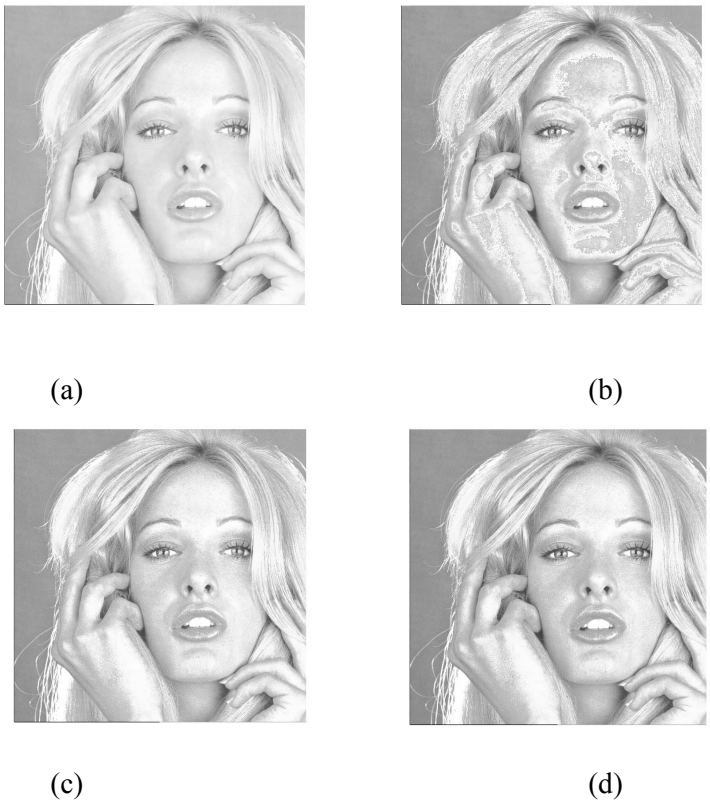
(g) (h) (i)

Figure 6: Comparisons of visual perception and embedding capacity of marked images for Lena, Barbara and Goldhill when the PEE-based RDH method is used in the first step. (a) Lena. (b) 10 pairs: 28.353 dB, 228,330 bits, RCE=0.5298. (c) 15 pairs: 25.2523 dB, 244,970 bits, RCE=0.5461. (d) Barbara. (e) 10 pairs: 29.542 dB, 123,879 bits, RCE=0.53068. (f) 16 pairs: 25.679 dB, 174,076 bits, RCE=0.54859. (g) Goldhill. (h) 10 pairs: 30.247 dB, 122,461 bits, RCE=0.5253. (i) 17 pairs: 25.438 dB, 176,616 bits, RCE=0.54753

Besides Lena, we also utilize two other images, i.e., Goldhill and Barbara, to further illustrate the feasibility of our proposed framework. The reason of selecting Barbara and Goldhill is that they are used respectively in Gao et al. [Gao and Shi (2015)] and Wu et al. [Wu, Dugelay and Shi (2015)]. From Figs. 5 and 6, one can observe that two RDH methods (i.e., the IT-based and PEE-based RDH methods) provide a large part of the payload and simultaneously maintain the visual quality as much as possible. Therefore, the modified image in the first step can be used again in the second step to achieve

contrast enhancement, and meanwhile, the rest payload is obtained. After two steps, the marked image with high PSNR value and good visual quality are generated. Similarly, two RDH methods are used for Goldhill to achieve high EC and maintain good visual quality (refer to Figs. 5 and 6 for details).

The experimental comparisons between Wu's [Wu, Dugelay and Shi (2015)] and two used RDH methods are shown in Fig. 7. Fig. 7(b) shows that when 20 peak pairs are modified for carrying data, Wu's method leads to the obvious visual distortion for Tiffany. However, two used RDH methods achieves better visual quality and larger EC compared with Wu's method (see Figs. 7(c) and 7(d) for details). For the IT-based method, when its block size is set 2×2, the maximum embedding rate approaches to 0.75 bpp. In contrast, the PEE-based RDH method generates prediction in a pixel-by-pixel manner, so it can achieve the embedding rate of close to 1 bpp. This is the reason that with the embedding rates increased, the PEE-based method can achieve larger EC and better visual quality than the IT-based one. When multiple-embedding is utilized in the IT-based method, the embedding performance can be increased.



(a) (b)

(c) (d)

Figure 7: Comparisons of visual perception and embedding capacity of marked image among Wu et al.'s [Wu, Dugelay and Shi (2015)] and the IT-based RDH method for Tiffany. (a) Original image. (b) [Wu, Dugelay and Shi (2015)], 20 pairs: 22.7509 dB, 177,158 bits, RCE=0.52758. (c) PEE-based RDH method, 20 pairs: 24.309 dB, 277,453 bits. (d) IT-based RDH method, 25 pairs: 22.239 dB, 245,281 bits, RCE=0.53399

Table 1: Comparison of amounts of embedded bits and PSNR among Gao et al. [Gao and Shi (2015); Wu, Dugelay and Shi (2015)], PEE-based RDH method, and IT-based RDH method

	Lena	Baboon	Airplane	Barbara	Tiffany	Boat
Gao	214,918 bits 25 dB	173,807 bits	353,310 bits	161,933 bits 25 dB	267,629 bits	221,558 bits
Wu	90,440 bits 24.91 dB	93,610 bits 24.34 dB	205,110 bits 23.71 dB	86,697 bits 24.36 dB	177,158 bits 22.75 dB	151,770 bits 23.28 dB
IT-based RDH method	228,930 bits 25.22 dB	174,018 bits 24.51 dB	359,575 bits 23.10 dB	167,826 bits 25.79 dB	245,281 bits 22.23 dB	191,425 bits 28.44 dB
PEE-based RDH method	228,330 bits 28.35 dB	177,219 bits 24.10 dB	391,357 bits 23.77 dB	174,076 bits 25.67 dB	277,453 bits 24.43 dB	245,117 bits 27.06 dB

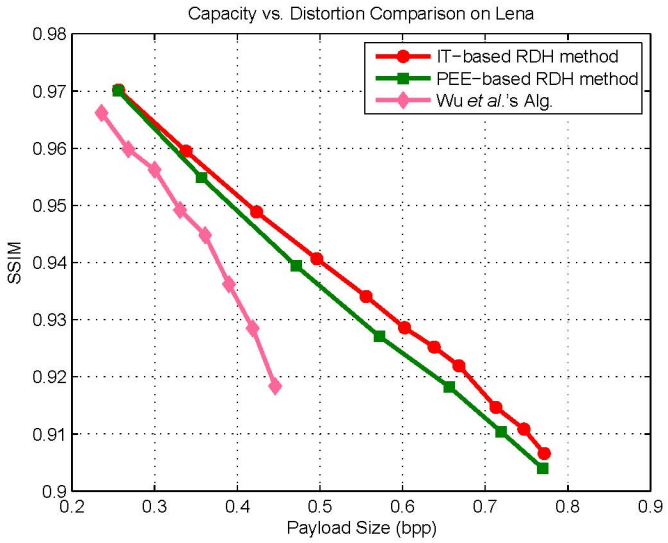


Figure 8: SSIM comparisons between IT-based RDH, PEE-based RDH and Wu's methods for Lena

Comparisons of the amount of embedded bits and PSNR values among Gao et al. [Gao and Shi (2015); Wu, Dugelay and Shi (2015)], PEE-based RDH method, and IT-based RDH method are listed in Tab. 1. Since we do not obtain the source code of Gao's method, we cannot enumerate the PSNR value of each image at the given EC. From Tab. 1, one can observe that two used methods can achieve higher visual quality and larger EC than Wu's and Gao's methods. It is clear that two used methods can embed more data while keeping higher visual quality.

Table 2: Performance comparison in terms of PSNR (in dB) and embedding capacity (EC, in bpp), and RCE on Kodak image database for $N_p = 10$

	PEE-based RDH			Wu's method		
Image	PSNR	Capacity	RCE	PSNR	Capacity	RCE
kodim01	31.469	0.307	0.523	31.406	0.282	0.524
kodim02	29.457	1.136	0.523	29.251	0.984	0.525
kodim03	29.2230	1.0294	0.5256	29.207	0.354	0.528
kodim04	29.2379	0.42449	0.5262	29.439	0.280	0.528
kodim05	29.8072	0.13344	0.5661	39.32	0.04216	0.508
kodim06	33.3125	0.62665	0.5090	33.997	0.276	0.510
kodim07	29.237	0.475	0.527	28.648	0.369	0.528
kodim08	32.5070	0.25045	0.5142	29.972	0.142	0.5249
kodim09	28.935	0.69911	0.5285	29.363	0.359	0.529
kodim10	32.233	0.61071	0.5143	28.535	0.339	0.529
kodim11	29.541	0.839	0.522	29.696	0.459	0.544
kodim12	29.0321	0.6491	0.5266	28.982	0.402	0.52743
kodim13	31.3441	0.29504	0.5144	28.967	0.203	0.52624
kodim14	29.823	0.359	0.528	29.412	0.214	0.52945
kodim16	30.283	0.585	0.526	29.764	0.281	0.529
kodim17	28.3282	0.79338	0.507	29.480	0.246	0.516
kodim18	32.6549	0.22099	0.5127	29.929	0.11892	0.5252
kodim19	28.9344	0.6651	0.5302	28.974	0.253	0.532
kodim20	30.1191	0.76441	0.5123	27.065	0.672	0.507
kodim21	28.9298	0.5496	0.5271	28.982	0.401	0.528
kodim22	29.3284	0.42422	0.5282	29.603	0.283	0.529
kodim23	28.9937	0.62547	0.5263	28.977	0.301	0.526
kodim24	32.1750	0.57101	0.5127	29.474	0.212	0.520

Table 3: Performance comparison between the PEE-based RDH and Wu's methods in terms of PSNR (in dB) and embedding capacity (EC, in bpp), and RCE on Lena for different N_p

	PEE-based RDH			Wu's method		
N_p	PSNR	Capacity	RCE	PSNR	Capacity	RCE
10	28.353	0.8710	0.5298	28.3202	0.201	0.53257
15	25.2523	0.93447	0.5461	25.6981	0.284	0.54593
20	23.038	1.027	0.5613	24.4067	0.361	0.55602

Another commonly-used quality measure, the SSIM index (structural similarity index), is also exploited in experiments for performance comparison. The SSIM index closer to 1 implies the similarity of two images is higher. Fig. 8 illustrates SSIM comparisons among the IT-based RDH, PEE-based RDH and Wu's methods. From Fig. 8, it can be observed that SSIM index of IT-based RDH method (or PEE-based RDH method) is higher than that of Wu's one at almost all ERs.

In addition, we have compared the PEE-based RDH method and Wu's method on Kodak

image database, which is composed of 24 color images with size of 512×768 or 768×512. For convenience of comparison, all color images are transformed into the gray-scale version. Since there exist a large amount of pixel on both ends of the histogram of kodim 15, the location map recording the pixels with overflow or underflow cannot be compressed efficiently. As a result, the obtained payload is very low for both the PEE-based RDH and Wu's methods. This is the reason that we did not illustrate the experimental results of kodim 15. From Tab. 2, one can observe that the PEE-based RDH method is superior to Wu's one for 23 images from Kodak image database.

Since we cannot obtain the source code of Gao's method, we only provide the comparison of RCE under different N_p for three methods (i.e., the IT-based RDH, PEE-based RDH and Wu's methods). And, two tables have been used to illustrate the comparison of RCE under different N_p among three methods for Lena. From Tabs. 3 to 4, it is observed that our proposed method can achieve better embedding performance than Wu's one. More importantly, our RCE is almost the same as that of Wu's method.

Table 4: Performance comparison between the IT-based RDH and Wu's methods in terms of PSNR (in dB) and embedding capacity (EC, in bpp), and RCE on Lena for different N_p

	IT-based RDH			Wu's method		
N_p	PSNR	Capacity	RCE	PSNR	Capacity	RCE
10	28.3417	0.72256	0.5298	28.3202	0.201	0.53257
15	25.229	0.87328	0.5460	25.6981	0.284	0.54593
20	23.1274	0.94606	0.5610	24.4067	0.361	0.55602

5 Conclusion

In this paper, we present a general framework to construct RDH with controlled contrast enhancement. According to our framework, any RDH method which can achieve high EC while maintain satisfactory visual quality, can be used in the first step. This implies the generalization of our proposed framework. Since the first step preserves satisfactory visual quality, I_W can be used in second step to increase contrast enhancement. Experimental results also demonstrate our proposed framework can achieve a better performance compared with Wu's and Gao's methods. However, in the experiments, we simply set $pT_h = 2vT_h$ and select N_p separately. In fact, under the given EC and the fixed degree of contrast enhancement, there are some combinations of pT_h, vT_h and N_p. In future, our work is to search the optimal combination which can achieve the highest EC and best visual quality.

Acknowledgement: This work was supported in part by National NSF of China (Nos. 61872095, 61872128, 61571139 and 61201393), New Star of Pearl River on Science and Technology of Guangzhou (No. 2014J2200085), the Open Project Program of Shenzhen Key Laboratory of Media Security (Grant No. ML-2018-03), the Opening Project of

Guang Dong Province Key Laboratory of Information Security Technology (Grant No. 2017B030314131-15), Natural Science Foundation of Xizang (No. 2016ZR-MZ-01).

References

Alattar, A. M. (2004): Reversible watermark using the difference expansion of a generalized integer transform. *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147-1156.

Celik, M. U.; Sharma, G.; Tekalp, A. M.; Saber, E. (2005): Lossless generalized-lsb data embedding. *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 157-160.

Chen, Y. Y.; Yin, B. X.; He, H. J.; Yan, S.; Chen, F. et al. (2018): Reversible data hiding in classification-scrambling encrypted-image based on iterative recovery. *Computers, Materials & Continua*, vol. 56, no. 2, pp. 299-312.

Fridrich, J.; Goljan, M.; Du, R. (2002): Lossless data embedding-new paradigm in digital watermarking. *Eurasip Journal on Applied Signal Processing*, vol. 2002, pp. 185-196.

Gao, G. Y.; Shi, Y. Q. (2015): Reversible data hiding using controlled contrast enhancement and integer wavelet transform. *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 2078-2082.

Gao, M.; Wang, L. (2013): Comprehensive evaluation for he based contrast enhancement. *Advances in Intelligent Systems and Applications*, vol. 2, pp. 331-338.

Hong, W.; Chen, T.; Wu, M. (2013): An improved human visual system based reversible data hiding method using adaptive histogram modification. *Optics Communications*, vol. 291, pp. 87-97.

Li, X. L.; Li, B.; Yang, B.; Zeng, T. Y. (2013): General framework to histogram-shifting-based reversible data hiding. *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2181-2191.

Li, X. L.; Yang, B.; Zeng, T. Y. (2011): Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3524-3533.

Ni, Z.; Shi, Y. Q.; Ansari, N.; Su, W. (2006): Reversible data hiding. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, pp. 354-362.

Sachnev, V.; Kim, H. J.; Nam, J.; Suresh, S.; Shi, Y. Q. (2009): Reversible watermarking algorithm using sorting and prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 989-999.

Tian, J. (2003): Reversible data embedding using a difference expansion. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890-896.

Wang, C.; Li, X. L.; Yang, B. (2010): High capacity reversible image watermarking based on in transform. *Proceedings of IEEE International Conference on Image Processing*, pp. 217-220.

Wang, X.; Li, X. L.; Yang, B.; Guo, Z. M. (2010): Efficient generalized integer transform for reversible watermarking. *IEEE Signal Processing Letters*, vol. 17, no. 6, pp. 567-570.

Weng, S.; Pan, J. S. (2016): Integer transform based reversible watermarking incorporating block selection. *Journal of Visual Communication and Image Representation*, vol. 35, pp. 25-35.

Wu, H. T.; Dugelay, J. L.; Shi, Y. Q. (2015): Reversible image data hiding with contrast enhancement. *IEEE Signal Processing Letters*, vol. 22, no. 1, pp. 81-85.

Wu, X.; Memon, N. (1997): Context-based, adaptive, lossless image coding. *IEEE Transactions on Communications*, vol. 45, no. 4, pp. 437-444.

Xiong, L. Z.; Qing, S. Y. (2018): On the privacy-preserving outsourcing scheme of reversible data hiding over encrypted image data in cloud computing. *Computers, Materials & Continua*, vol. 55, no. 3, pp. 523-539.

Xuan, G. R.; Yang, C. Y.; Zhen, Y. Z.; Shi, Y. Q. (2004): Reversible data hiding using integer wavelet transform and companding technique. *Proceedings of International Workshop on Digital Forensics and Watermarking*, vol. 5, pp. 23-26.

Zhang, X. P. (2011): Reversible data hiding in encrypted images. *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255-258.