PATCH-LEVEL SELECTION AND BREADTH-FIRST PREDICTION STRATEGY FOR REVERSIBLE DATA HIDING

Hanzhou Wu

Shanghai University, Shanghai 200444, China

ABSTRACT

A core work in reversible data hiding is designing an embedding method enabling the hider to take advantages of smooth elements as many as possible while the detection procedure for marked elements is invertible to the receiver. It motivates us to introduce a novel patch-level selection and breadth-first prediction strategy for efficient reversible data hiding. However, different from conventional works, the proposed work allows us to preferentially and simultaneously use adjacent smooth elements as many as possible. Experiments show that it significantly outperforms a part of state-of-the-arts at relatively low embedding rates, demonstrating the superiority.

Index Terms— reversible data hiding (RDH), watermarking, breadth-first search (BFS), patch, fragile.

1. INTRODUCTION

People are experiencing the many advantages of multimedia technology and network services, among which cloud storage has become a popular service. The cloud storage service allows users to store media files by an outsourced manner. A cloud manager may insert extra data to a media file for content authentication. Since the manager cannot permanently distort the content, the insertion operation should be erasable, which gives reversible data hiding (RDH) [1, 2, 3] life in this special area since RDH ensures that both the hidden data and the original media content can be perfectly reconstructed. RDH is a fragile technique, meaning that, when a media file containing extra data was manipulated, one will find it is not authentic and the original media content may be not fully reconstructed. RDH has also been applied to other sensitive applications requiring no degradation of the cover such as military imagery.

A straightforward idea [3, 4] for RDH is losslessly compression. The compression procedure often executes on the LSBs of pixels, where the compression ratio is relatively low, which does not provide a sufficient payload. More techniques such as histogram shifting (HS) [1] and difference expansion (DE) [2] have been proposed to achieve a higher embeddable payload or a lower distortion, e.g., many advanced RDH systems [5, 6, 7, 8, 9, 10] use HS or its variants such as prediction error (PE) [5] and prediction error expansion (PEE) [9, 10].

A RDH system can be evaluated by the payload-distortion performance [11, 12]. Namely, for a payload, it is required to keep distortion as low as possible. In other words, for a fixed distortion level, it is desirable to embed as many message bits as possible. Strong spatial correlations exist between neighboring pixels in an image. It inspires researchers to design prediction based RDH algorithms, which use the prediction-errors (PEs) to carry a payload. A common operation [7, 11] used in RDH is therefore to generate a prediction error histogram (PEH) for data embedding. On the one hand, the PEs are noise-like components of the cover, meaning that, the slight modification to PEs would not result in obvious artifacts. On the other hand, as a pooled vector, the PEH allows us to embed secret data in an effective way. It indicates that, to provide superior performance, the design of a RDH system may focus on PEH generation and its modification. Lots of works [19, 18, 17, 16, 15, 13, 14] are developed along this line.

In this paper, we present a new pixel selection and prediction strategy to PEH generation for efficient RDH. Different from many previous works, our method allows a hider to preferentially and simultaneously use adjacent smooth elements as many as possible, which can benefit data embedding procedure a lot. Our experiments show that the proposed work ensures reversibility and significantly outperforms related works in terms of payload-distortion performance.

The rest of this paper is organized as follows. We describe the proposed method in Section 2, followed by experiments in Section 3. Finally, we conclude this paper in Section 4.

2. PROPOSED METHOD

With an image $\mathbf{x} = \{x_{i,j}\}^{h\times w}$, $x_{i,j} \in \mathcal{I} = \{0, 1, ..., 2^d - 1\}$, $d > 0$, the proposed work embeds a payload $\mathbf{m} = \{m_i\}_{i=1}^{l} \subset \{0, 1\}^l$, into $\mathbf{x}$, resulting in a marked image $\mathbf{y} = \{y_{i,j}\}^{h\times w}$, $y_{i,j} \in \mathcal{I}$. A patch is defined as a small matrix sized $r \times c$, where $1 \leq r \leq h$, $1 \leq c \leq w$. The proposed work first collects a number of disjoint patches from $\mathbf{x}$, each of which is then classified as "a smooth patch" or "a complex patch". Only smooth patches are used for pixel prediction and data embedding. The complex patches are unchanged. Then, by

It was supported by National Natural Science Foundation of China under Grant No. 61902235, "Chen Guang" project funded by Shanghai Municipal Education Commission and Shanghai Education Development Foundation.

using proposed breadth-first prediction procedure, the pixels to be potentially embedded in smooth patches are predicted, which allows a data hider to collect all required PEs. By computing the local-complexity of each pixel to be potentially embedded, an ordered PE sequence can be generated. Thereafter, with optimized HS parameters, $\mathbf{m}$ can be successfully embedded into the corresponding PEH, finally resulting in a marked image $\mathbf{y}$. Notice that, to avoid the underflow and overflow problem during data embedding, a losslessly compressed location map has also been self-embedded together with $\mathbf{m}$.

At the data receiver side, according to the necessary parameters extracted from $\mathbf{y}$, one can successfully identify the marked pixels and determine the marked PE sequence. Thus, with the HS parameters extracted from $\mathbf{y}$, $\mathbf{m}$ can be perfectly reconstructed. Meanwhile, the original image $\mathbf{x}$ can be also recovered according to the extracted location map and other necessary information. We detail each part below.

2.1. Patch-Level Selection

We divide $\mathbf{x}$ into two disjoint subsets, denoted by S_A and S_B. Both S_A and S_B are further divided into two disjoint subsets, i.e., $S_A^{(0)}$, $S_A^{(1)}$, $S_B^{(0)}$, and $S_B^{(1)}$. We have $\mathbf{x} = S_A \cup S_B$, $S_A = S_A^{(0)} \cup S_A^{(1)}$, and $S_B = S_B^{(0)} \cup S_B^{(1)}$. $S_A^{(0)}$ will be unchanged and used for predicting the pixels in $S_A^{(1)}$. The pixels in $S_A^{(1)}$ will carry $\mathbf{m}$. $S_B^{(0)}$ will be unchanged, and $S_B^{(1)}$ will be used for storing the secret key and data embedding parameters. For the sake of simplicity, in default, we set $S_B^{(1)}$, i.e., $S_B^{(1)} = \{x_{i,j} | i = h, 1 \le j \le w\}$. Notice that, one may redefine $S_B^{(1)}$.

With patch size, by processing $\mathbf{x}$ from left to right and from top to bottom, we can collect $N_p = \lfloor \frac{h-1}{r} \rfloor \times \lfloor \frac{w}{c} \rfloor$ patches. Let $\mathbf{p}_k = \{p_{i,j}^{(k)}\}^{r \times c}$ be the k-th patch indexed by a row-by-row manner. For each $\mathbf{p}_k$, we randomly generate a binary matrix $\mathbf{b}_k = \{b_{i,j}^{(k)}\}^{r \times c} = \{0,1\}^{r \times c}$ by a secret key. And, $\sum_{i=1}^{r} \sum_{j=1}^{c} b_{i,j}^{(k)} = \alpha^{(k)}$, where $\alpha^{(k)} \ge 1$ is a predetermined integer parameter. For each $\mathbf{p}_k$, we determine a set $\mathbf{s}_k$ including such elements that they are corresponding to 1 in $\mathbf{b}_k$:

$$\mathbf{s}_k = \{p_{i,j}^{(k)} \mid 1 \le i \le r, 1 \le j \le c, b_{i,j}^{(k)} = 1\}. \quad (1)$$

For $\mathbf{s}_k$, we compute $d^{(k)} = \max\{s | s \in \mathbf{s}_k\} - \min\{s | s \in \mathbf{s}_k\}$. We classify $\mathbf{p}_k$ as a smooth patch if $d^{(k)} \le \beta^{(k)}$. Otherwise, $\mathbf{p}_k$ is considered as a complex patch. Here, $\beta^{(k)} \ge 0$ is also a predetermined integer parameter. A smooth patch implies that, it would be suitable for data embedding. A complex patch will not be embedded. Thus, we determine S_A as:

$$S_A = \bigcup_{d^{(k)} \le \beta^{(k)}} \mathbf{p}_k. \quad (2)$$

Then, $S_A^{(0)}$ and $S_A^{(1)}$ are determined as:

$$\begin{cases} S_A^{(0)} = \bigcup_{d^{(k)} \le \beta^{(k)}} \mathbf{s}_k, \\ S_A^{(1)} = \bigcup_{d^{(k)} \le \beta^{(k)}} \mathbf{p}_k \setminus \mathbf{s}_k. \end{cases} \quad (3)$$

Furthermore, S_B and $S_B^{(0)}$ are determined as:

$$\begin{cases} S_B = \mathbf{x} \setminus S_A, \\ S_B^{(0)} = S_B \setminus S_B^{(1)}. \end{cases} \quad (4)$$

It is seen that, the patch-level selection procedure generates four disjoint pixel-sets depending on a sequence of parameters $\{\alpha^{(1)}, ..., \alpha^{(N_p)}\}$ and $\{\beta^{(1)}, ..., \beta^{(N_p)}\}$. In default, we will use $\alpha^{(1)} = ... = \alpha^{(N_p)} = \alpha$ and $\beta^{(1)} = ... = \beta^{(N_p)} = \beta$. Thus, we only set two parameters.

2.2. Breadth-First Prediction

Once we collect four sets, we predict the pixels in $S_A^{(1)}$ and collect PEs for data embedding. We propose a breadth-first prediction algorithm to determine the prediction values of pixels in $S_A^{(1)}$ by $S_A^{(0)}$. It corresponds to an iterative process. During the prediction process, we will select a specific pixel from S_A. And, we will predict all its adjacent non-predicted pixels belonging to $S_A^{(1)}$ simultaneously. The "breadth" prediction process will be terminated when all required pixels are predicted. This "breadth-first" perspective requires a low computational cost, i.e., $O(4|S_A|)$. We detail it below.

Let $\hat{x}_{i,j}$ be the prediction value of $x_{i,j}$. We first append all pixels in $S_A^{(0)}$ to an empty queue Q, and mark pixels in $S_A^{(0)}$ as *processed*. Then, iteratively, we select a pixel $x_{i,j}$ from Q and perform the following three steps until Q is empty.

Step 1. Collect all the adjacent pixels of $x_{i,j}$. The pixels should belong to $S_A^{(1)}$ and have not been marked as *processed*. A pixel $x_{u,v}$ is adjacent to $x_{i,j}$ if $|u-i| + |v-j| = 1$.

Step 2. For each collected pixel $x_{u,v}$ (if any), if $x_{i,j} \in S_A^{(0)}$, we predict $x_{u,v}$ by $\hat{x}_{u,v} = x_{i,j}$, otherwise, we must have $x_{i,j} \in S_A^{(1)}$ and predict $x_{u,v}$ by $\hat{x}_{u,v} = \hat{x}_{i,j}$.

Step 3. Append all collected pixels (if any) to Q, and remove $x_{i,j}$ from Q. Mark the collected pixels as *processed*.

The prediction algorithm ensures that, the prediction value of a pixel in $S_A^{(1)}$ is predicted from its nearest pixel with Manhattan distance that belongs to $S_A^{(0)}$. Obviously, the Manhattan distance is at least 1. And, its upper bound is $r + c$. When we keep $r + c$ low, the prediction accuracy will be satisfactory. It relies on the assumption that strong spatial correlations exist between pixels with a small distance.

2.3. Data Embedding

After predicting pixels in $S_A^{(1)}$, we can determine the PEs of pixels. For each $x_{i,j} \in S_A^{(1)}$, the PE is $e_{i,j} = x_{i,j} - \hat{x}_{i,j}$.

We sort the PEs so that small PEs can be used first. We use the prediction values of pixels in $S_A^{(1)}$ and the original values of pixels in $S_A^{(0)}$ to construct a local-complexity evaluation function. For each $x_{i,j} \in S_A^{(1)}$, we determine

$$N(x_{i,j}) = \{x_{i',j'} \in S_A | \, |i - i'| + |j - j'| \le 1\}. \quad (5)$$

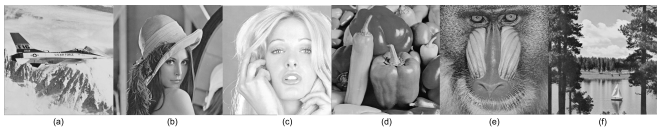

Fig. 1. Six standard images sized $512 \times 512 \times 8$: (a) *Airplane*, (b) *Lena*, (c) *Tiffany*, (d) *Peppers*, (e) *Baboon*, and (f) *Boat*.

The local-complexity for $x_{i,j}$ is then computed as:

$$\rho_{i,j} = \max\{x^{\circ}_{u,v} \in N(x_{i,j})\} - \min\{x^{\circ}_{u,v} \in N(x_{i,j})\}, \quad (6)$$

where

$$x^{\circ}_{u,v} = \begin{cases} x_{u,v}, \text{if } x_{u,v} \notin S_A^{(0)}, \\ \hat{x}_{u,v}, \text{if } x_{u,v} \in S_A^{(1)}. \end{cases} \quad (7)$$

A smaller local-complexity implies better prediction accuracy. Accordingly, we can sort all PEs in an increasing order of the local-complexity so that small PEs can be embedded first. We use $\mathbf{e} = \{e_i\}_{i=1}^{n_e}$ to represent the sorted PE sequence. **m** will be embedded into **e** by shifting HS bins. Let (l_p, r_p, T_d) denote the shifting parameters, where $l_p < r_p$ and $T_d > 0$. For a bit $b \in \{0, 1\}$ and $e_i \in \mathbf{e}$, the marked PE $\hat{e}_i$ is:

$$\hat{e}_i = \begin{cases} e_i - (l_p - e_i) - b, & \text{if } l_p - T_d < e_i \le l_p, \\ e_i + (e_i - r_p) + b, & \text{if } r_p \le e_i < r_p + T_d, \\ e_i + T_d, & \text{if } e_i \ge r_p + T_d, \\ e_i - T_d, & \text{if } e_i \le l_p - T_d, \\ e_i, & \text{otherwise.} \end{cases} \quad (8)$$

$\hat{e}_i$ will be added to the prediction value of the corresponding pixel so as to generate the marked pixel. It is mentioned that, we will terminate the data embedding procedure when **m** is fully embedded. It means that, there is a PE position $t \le n_e$ that all PEs $\{e_i \mid i > t\}$ are unchanged. It is necessary to optimize (l_p, r_p, T_d) such that the distortion by embedding **m** into $\{e_i\}_{i=1}^{t}$ can be kept low. For a specific t, we can easily determine the corresponding PEH, i.e.,

$$h(k) = |\{e_i = k \mid 1 \le i \le t\}|. \quad (9)$$

To hide **m**, we have $\sum_{v \in (l_p - T_d, l_p] \cup [r_p, r_p + T_d)} h(v) = l$, where l is the length of **m**. Moreover, $e_t \in (l_p - T_d, l_p] \cup [r_p, r_p + T_d)$. We estimate the embedding distortion by

$$\begin{aligned} D_t(l_p, r_p, T_d) = & \left[\sum_{v \le l_p - T_d} h(v) + \sum_{v \ge r_p + T_d} h(v) \right] \cdot T_d^2 \\ & + \sum_{k \in [0, T_d)} \left[\frac{h(l_p - k)}{2} + \frac{h(r_p + k)}{2} \right] \cdot k^2 \\ & + \sum_{k \in [0, T_d)} \left[\frac{h(l_p - k)}{2} + \frac{h(r_p + k)}{2} \right] \cdot (k+1)^2. \end{aligned} \quad (10)$$

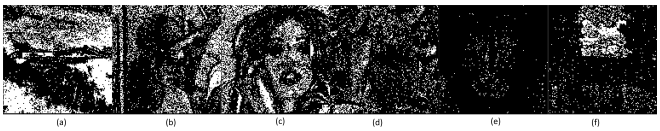

Fig. 2. The resulting distribution maps by using patch-level selection procedure: (a) *Airplane*, (b) *Lena*, (c) *Tiffany*, (d) *Peppers*, (e) *Baboon*, (f) *Boat*. Here, $r = c = \alpha = \beta = 4$. The white pixels belong to S_A and black for S_B.

We call t usable if there exists such (l_p, r_p, T_d) meeting that $\sum_{v \in (l_p - T_d, l_p] \cup [r_p, r_p + T_d)} h(v) = l$ and $e_t \in (l_p - T_d, l_p] \cup [r_p, r_p + T_d)$. Accordingly, we can collect all usable t, which can be done during the process of orderly collecting PEs. For each usable t, we can further determine the suboptimal (l_p, r_p, T_d) by minimizing Eq. (10). Then, we can get global-optimal (l_p, r_p, T_d). In this way, **m** can be embedded.

2.4. Parameter Embedding and Extraction

For reversibility, the secret key, r, c, α, β, and (l_p, r_p, T_d) will be embedded into the LSBs of some pixels belonging to $S_B^{(1)}$ (by LSB replacement), the original LSBs of which are considered as a part of **m**. We use 16 bits to store the secret key. And, r (4 bits), c (4 bits), α (6 bits) and β (4 bits) are stored with 18 bits. $3d$ bits are used to store (l_p, r_p, T_d). In addition, before data embedding, a part of boundary pixels in $S_A^{(1)}$ are adjusted into the range $[T_d, 2^d - 1 - T_d]$ and recorded to constitute a location map, which will be losslessly compressed by arithmetic coding. The compressed location map is also considered as a part of **m**. Thus, the size of pure payload is $L = l - 34 - 3d - C_{\text{LM}}$, where C_{LM} is the size of losslessly compressed location map. In most cases, C_{LM} for a natural image could be quite small, meaning that, its impact on L can be roughly ignored. In experiments, we use l. For a receiver, he should extract the parameters, which is straightforward.

2.5. Data Extraction and Image Recovery

By extracting (r, c), a receiver collects all N_p patches. The N_p matrices are computed by the key and α. With β and the matrices, the receiver can identify smooth patches. Accordingly, $S_A^{(0)}$, $S_A^{(1)}$, $S_B^{(0)}$ and $S_B^{(1)}$ can be perfectly reconstructed.

Since $S_A^{(0)}$ is unchanged, the receiver can reconstruct the prediction values of pixels in $S_A^{(1)}$. And, the corresponding marked PEs can be all collected. As the local-complexity of a pixel only relies on the prediction values of pixels in $S_A^{(1)}$ and the original values of pixels in $S_A^{(0)}$, the receiver can generate a marked PE sequence identical to the sender side. Thus, the receiver can extract **m** without error. For a marked PE $\hat{e}_i$, the secret bit can be easily extracted in an inverse way. The original PE e_i can be easily determined out as well. e_i will be added to the prediction value of the corresponding pixel

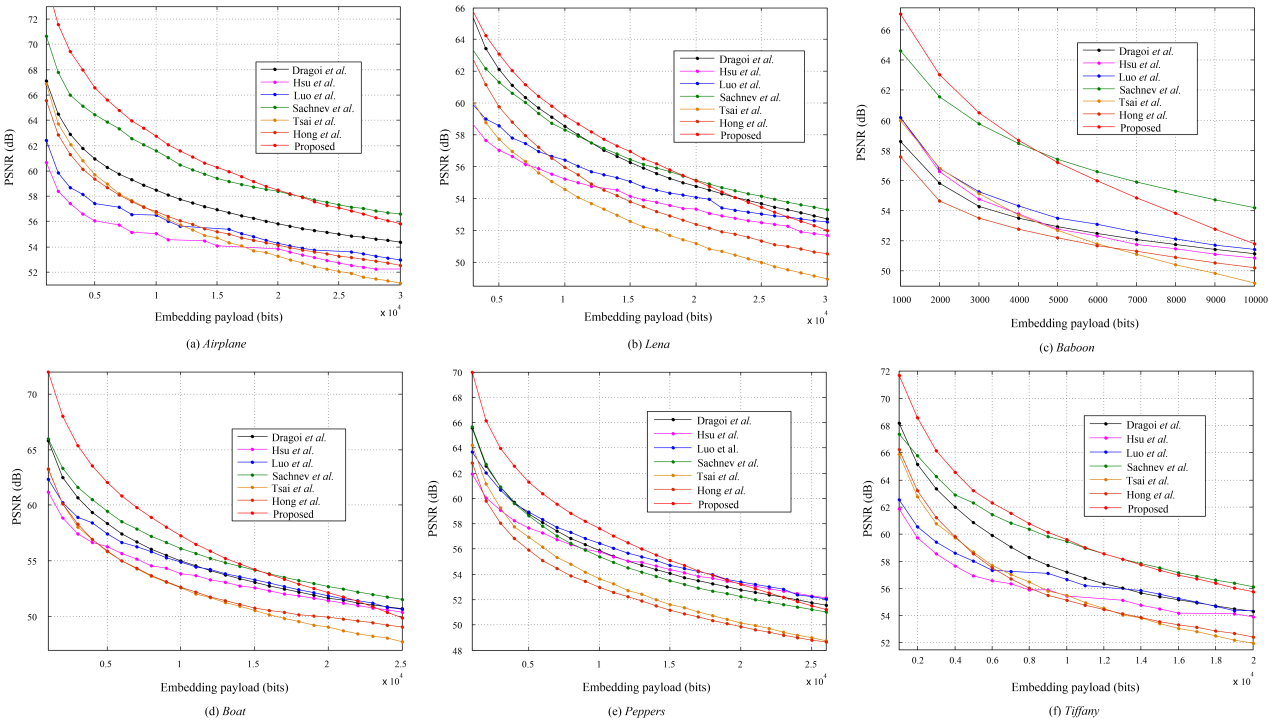

Fig. 3. Performance comparison between proposed work and the methods introduced by Dragoi *et al*. [14], Hsu *et al*. [20], Luo *et al*. [16], Sachnev *et al*. [10], Tsai *et al*. [5] and Hong *et al*. [13].

to reconstruct the preprocessed pixel value at the sender side. With **m**, the receiver can further obtain the original LSBs of specific pixels in $S_B^{(1)}$ as well as the losslessly compressed location map, which allows **x** to be perfectly recovered.

3. EXPERIMENTAL RESULTS

We take *Airplane*, *Lena*, *Tiffany*, *Peppers*, *Baboon*, *Boat* in Fig. 1 varying from smooth to complex for experiments. We apply proposed patch-level selection procedure to the images. Fig. 2 shows the distribution maps. It is seen that, the proposed work selects smooth content from an image. The selected pixels to be potentially embedded could be adjacent to each other. It implies that, many adjacent smooth pixels can be simultaneously embedded, having the potential to provide superior performance. To evaluate the payload-distortion performance, we compare the proposed work with a part of advanced HS based RDH works introduced by Dragoi *et al*. [14], Hsu *et al*. [20], Luo *et al*. [16], Sachnev *et al*. [10], Tsai *et al*. [5] and Hong *et al*. [13]. In our experiments, we vary α from 1 to $\lfloor \frac{r \times c}{2} \rfloor$, and β from 0 to 8 for all images. The (α, β) resulting in the best performance will be chosen for RDH. We vary both r and c from 1 to 8. It is required that $r \times c \geq 2$. And, we use $2\alpha \leq r \times c$, which ensures that $|S_A^{(0)}| \leq |S_A^{(1)}|$ so as to provide a sufficient payload. Notice that, the parameter optimization is only available to a data hider.

As shown in Fig. 3, the PSNR gain is significant at relatively low embedding rates for all test images. For *Baboon*, the performance decline significantly. The reason is that, the proposed work aims to use smooth content. However, due to between-image difference, different images have different smoothness, meaning that, different images result in different performance. Moreover, the proposed pixel prediction algorithm is quite effective for smooth images since in smooth images, the values of adjacent pixels are quite close to each other. However, for a complex image, e.g., *Baboon*, the difference between two adjacent pixels could be very high, meaning that, the number of smooth pixels is limited, which cannot benefit the proposed prediction procedure. From the viewpoint of real-world, more natural images are likely to have relatively sufficient smooth pixels, indicating that, the proposed work has the potential to provide superior performance.

4. CONCLUSION

Improving the utilization of smooth contents has been a core topic in RDH. It motivates the authors to introduce a novel patch-level selection and breadth-first prediction strategy to further take advantages of smooth pixels as many as possible. Experiments have shown that, the proposed work provides superior performance at relatively low embedding rates compared to state-of-the-arts. A future work is to improve the performance for complex images at high embedding rates.

5. REFERENCES

[1] Z. Ni, Y. Shi, N. Ansari and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, 16(3): 354-362, Mar. 2006.

[2] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, 13(8): 890-896, Aug. 2003.

[3] M. U. Celik, G. Sharma and A. M. Tekalp, "Lossless watermarking for image authentication: a new framework and an implementation," *IEEE Trans. Image Process.*, 15(4): 1042-1049, Apr. 2006.

[4] J. Fridrich, M. Goljan and R. Du, "Invertible authentication," In: *Proc. SPIE*, San Jose, CA, vol. 3971, pp. 197-208, Aug. 2001.

[5] P. Tsai, Y. Hu and H. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Process.*, 89(6): 1129-1143, Jun. 2009.

[6] X. Li, B. Li, B. Yang and T. Zeng, "General framework to histogram-shifting-based reversible data hiding," *IEEE Trans. Image Process.*, 22(6): 2181-2191, Jun. 2013.

[7] H. Wu, H. Wang and Y. Shi, "PPE-based reversible data hiding," In: *ACM Workshop Inf. Hiding Multimed. Security*, pp. 187-188, Jun. 2016.

[8] X. Li, W. Zhang, X. Gui and B. Yang, "Efficient reversible data hiding based on multiple histogram modification," *IEEE Trans. Inf. Forensics Security*, 10(9): 2016-2027, Sept. 2015.

[9] X. Li, B. Yang and T. Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," *IEEE Trans. Image Process.*, 20(12): 3524-3533, Dec. 2011.

[10] V. Sachnev, H. Joong Kim, J. Nam, S. Suresh and Y. Q. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Trans. Circuits Syst. Video Technol.*, 19(7): 989-999, Jul. 2009.

[11] H. Wu, H. Wang and Y. Shi, "Dynamic content selection-and-prediction framework applied to reversible data hiding," In: *IEEE Int. Workshop Inf. Forensics Security*, pp. 1-6, Dec. 2016.

[12] B. Ma and Y. Shi, "A reversible data hiding scheme based on code division multiplexing," *IEEE Trans. Inf. Forensics Security*, 11(9): 1914-1927, Sept. 2016.

[13] W. Hong, T. S. Chen and C. W. Shiu, "Reversible data hiding for high quality images using modification of prediction errors," *J. Syst. Softw.*, 82(11): 1833-1842, Nov. 2009.

[14] I. Dragoi, D. Coltuc and I. Caciula, "Gradient based prediction for reversible watermarking by difference expansion," In: *Proc. ACM Workshop Inf. Hiding Multimed. Security*, pp. 35-40, Jun. 2014.

[15] C. Yang and M. Tsai, "Improving histogram-based reversible data hiding by interleaving prediction," *IET Image Process.*, 4(4): 223-234, Aug. 2010.

[16] L. Luo, Z. Chen, M. Chen, X. Zeng and Z. Xiong, "Reversible image watermarking using interpolation technique," *IEEE Trans. Inf. Forensics Security*, 5(1): 187-193, Mar. 2010.

[17] D. M. Thodi and J. J. Rodrguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, 16(3): 721-730, Mar. 2007.

[18] I. Dragoi and D. Coltuc, "Local-prediction-based difference expansion reversible watermarking," *IEEE Trans. Image Process.*, 23(4): 1779-1790, Apr. 2014.

[19] B. Ou, X. Li, Y. Zhao, R. Ni and Y. Shi, "Pairwise prediction-error expansion for efficient reversible data hiding," *IEEE Trans. Image Process.*, 22(12): 5010-5021, Dec. 2013.

[20] F. Hsu, M. Wu and S. Wang, "Reversible data hiding using side-match prediction on steganographic images," *Multimed. Tools Appl.*, 67(3): 571-591, Dec. 2013.