IMPROVED PAIRWISE PIXEL-VALUE-ORDERING FOR HIGH-FIDELITY REVERSIBLE DATA HIDING

Ioan Catalin Dragoi, Ion Caciula and Dinu Coltuc

Faculty of Electrical Engineering, Electronics and Information Technology
Valahia University of Targoviste, Romania
Email: {catalin.dragoi, ion.caciula, dinu.coltuc}@valahia.ro

ABSTRACT

Pixel-value-ordering (PVO) appears as an efficient technique for high-fidelity reversible data hiding. This paper proposes a reversible data hiding scheme based on the pairwise PVO framework with improved difference equations. Both the pixel pair selection and the embedding algorithms are also streamlined. The proposed scheme uses a block classification approach based on a local complexity metric. Uniform blocks are processed using the proposed improved pairwise PVO algorithm. Slightly noisy blocks are embedded using a classic PVO scheme and noisy blocks are kept unchanged. The optimal embedding parameters for a given capacity are determined by linear programming. The proposed pairwise PVO approach outperforms other state-of-the-art schemes.

Index Terms— pixel-value-ordering, reversible data hiding, pairwise embedding, linear programming

1. INTRODUCTION

Reversible data hiding (RDH) embeds data into a digital host and ensures the exact recovery of both the original cover and the embedded message [1]. The standard approaches for RDH into clear images consist of prediction error expansion (PEE) and histogram bin shifting (PE-HS), see [2]. Pixel-value-ordering (PVO) appeared recently as an interesting and efficient alternative to the PE-HS embedding for very low distortion image RDH.

The basic PVO framework was introduced in [3]. As opposed to PE-HS/PEE, the PVO framework is block based. The pixels in each image block are sorted in ascending order and the hidden data is inserted by expanding the difference between the first two and last two elements of the sorted block. This is accomplished by shifting outwards the first and last sorted values, maintaining the same sorting order for the data extraction stage. The PVO difference equations of [3] were later refined in [4]. The number of embeddable pixels per block was also increased in [5]. Some PE-HS/PEE schemes were also derived from [3], notably [6, 7] which use the PVO difference equations as a predictor.

This paper proposes an improved PVO scheme inspired by the pairwise PVO of [5]. Compared with [5], the PVO difference equations are modified in order to increase the number of embedded pairs. Furthermore, a pair classification scheme simplifies the embedding equations. Finally, optimal parameters (thresholds and block size) are determined by using a linear programming model. The outline of the paper is as follows. The related work on PVO schemes is briefly discussed in Section 2. The proposed approach is introduced in Section 3. The experimental results are presented in Section 4 and the conclusions are drawn in Section 5.

2. RELATED WORK

The recently proposed PVO schemes either follow the embedding framework introduced in [3], or the PVO as a predictor approach of [6]. Since the proposed scheme belongs to the first category, we shall briefly describe the two most relevant reversible data hiding schemes that use this framework.

2.1. The IPVO scheme of Peng et al. [4]

The PVO embedding framework splits the host image into non-overlapping blocks of n pixels. For each block, the pixels $x_1, \ldots, x_n$ are sorted in ascending order based on their graylevel value. Let $x_{\sigma(1)} \leq x_{\sigma(2)} \leq ... \leq x_{\sigma(n)}$ be the sorted pixels. Two pixels are considered as possible hosts for the hidden data, namely $x_{\sigma(1)} = \min(x_1, x_2, ..., x_n)$ and $x_{\sigma(n)} = \max(x_1, x_2, ..., x_n)$. The remaining pixels are used to compute the complexity of the block. The blocks found too complex (based on a complexity threshold) are skipped. Two of the remaining pixels, $x_{\sigma(2)}$ and $x_{\sigma(n-1)}$, are also used as reference values for $x_{\sigma(1)}$ and $x_{\sigma(n)}$, respectively.

The position of the maximum/minimum in the block is then compared with the position of its reference value:

$$\begin{aligned} u_1 &= \min(\sigma(1), \sigma(2)) \quad & u_2 &= \min(\sigma(n-1), \sigma(n)) \\ v_1 &= \max(\sigma(1), \sigma(2)) \quad & v_2 &= \max(\sigma(n-1), \sigma(n)) \end{aligned} \tag{1}$$

This work was supported by UEFISCDI Romania, in the frame of PNIII-P4-IDPCE-2016-0339 and PN-III-P1-1.1-PD-2016-1666 Grants.

The algorithm then computes the differences between the selected pixels and their closest reference value:

$$d_l = x_{u_1} - x_{v_1} \quad d_r = x_{u_2} - x_{v_2} \tag{2}$$

The order of the difference therms is controlled by (1) (either *selected−reference* or *reference−selected*).

The PVO framework uses a fixed difference value to identify the host pixels (as opposed to PEE/HS schemes, where different prediction errors are selected based on the host image). The payload is controlled by the complexity threshold and the size of the blocks. In contrast with its counterpart in [3], equation (2) can produce negative values for d_l and d_r. This allows for the fixed difference pair $\{0, 1\}$ to be used for host selection. Note that the fixed difference from [3] is $\{1\}$, which is equivalent to $\{-1, 1\}$ when using (2). This is a key aspect of [4], since the "0" difference provides a larger embedding room than "-1".

The minimum of the block, $x_{\sigma(1)}$, is modified as follows:

$$x'_{\sigma(1)} = \begin{cases} x_{\sigma(1)} - w & \text{if } d_l \in \{0, 1\} \\ x_{\sigma(1)} - 1 & \text{if } d_l < 0 \text{ or } d_l > 1 \end{cases} \tag{3}$$

where $w \in \{0, 1\}$ is the data bit to be embedded. Similarly, for the maximum $x_{\sigma(n)}$ one has:

$$x'_{\sigma(n)} = \begin{cases} x_{\sigma(n)} + w & \text{if } d_r \in \{0, 1\} \\ x_{\sigma(n)} + 1 & \text{if } d_r < 0 \text{ or } d_r > 1 \end{cases} \tag{4}$$

The embedding process is then repeated for the next block, and so on.

Note that the sorting operation is necessary in order to determine d_l and d_r, but the pixels within the block are not actually moved. It should be also observed that by increasing $x_{\sigma(1)}$ by one graylevel, d_l either decreases by one (if $d_l = x_{\sigma(2)} - x_{\sigma(1)}$), or increases by one when $d_l = x_{\sigma(1)} - x_{\sigma(2)}$. This ensures the correct extraction of the hidden data (see [4] for a complete description of the decoding stage).

Equation (4) can be easily derived from (3). This is valid for all PVO based embedding equations. Thus, for the remaining PVO schemes, only the first equation of the pair will be provided.

2.2. The pairwise PVO scheme of Ou et al. [5]

Besides $x_{\sigma(1)}$ and $x_{\sigma(n)}$, the pairwise approach of Ou et al. [5] also considers $x_{\sigma(2)}$ and $x_{\sigma(n-1)}$ as possible host for the hidden data. Therefore $x_{\sigma(3)}$ and $x_{\sigma(n-2)}$ become the new reference values for $x_{\sigma(1)}$, $x_{\sigma(2)}$ and $x_{\sigma(n)}$, $x_{\sigma(n-1)}$, respectively:

$$\begin{aligned} d_{l1} &= x_{\sigma(3)} - x_{u_1} \quad & d_{r1} &= x_{u_2} - x_{\sigma(n-2)} \\ d_{l2} &= x_{\sigma(3)} - x_{v_1} \quad & d_{r2} &= x_{v_2} - x_{\sigma(n-2)} \end{aligned} \tag{5}$$

where u_1, v_1, u_2 and v_2 are computed with (1). Since $x_{\sigma(1)} \leq x_{\sigma(2)} \leq x_{\sigma(3)} \leq ... \leq x_{\sigma(n-2)} \leq x_{\sigma(n-1)} \leq x_{\sigma(n)}$, negative differences cannot be obtained.

The four possible hosts are processed as pairs: (x_{u_1}, x_{v_1}) and (x_{u_2}, x_{v_2}). The (x_{u_1}, x_{v_1}) pair is modified as follows:

$$(x'_{u_1}, x'_{v_1}) = \begin{cases} (x_{u_1}, x_{v_1}) - (w_1, w_2) & \text{if } d_{l1} = 1 \text{ and } d_{l2} = 1 \\ (x_{u_1}, x_{v_1}) - (w, w) & \text{if } d_{l1} = 2 \text{ and } d_{l2} = 2 \\ (x_{u_1}, x_{v_1}) - (w, 1) & \text{if } d_{l1} = 1 \text{ and } d_{l2} > 1 \\ (x_{u_1}, x_{v_1}) - (1, w) & \text{if } d_{l1} > 1 \text{ and } d_{l2} = 1 \\ (x_{u_1}, x_{v_1}) - (w, 0) & \text{if } d_{l1} = 0 \text{ and } d_{l2} = 0 \\ (x_{u_1}, x_{v_1}) - (0, w) & \text{if } d_{l1} = 0 \text{ and } d_{l2} = 1 \\ (x_{u_1}, x_{v_1}) - (1, 0) & \text{if } d_{l1} > 0 \text{ and } d_{l2} = 0 \\ (x_{u_1}, x_{v_1}) - (0, 1) & \text{if } d_{l1} = 0 \text{ and } d_{l2} > 1 \\ (x_{u_1}, x_{v_1}) - (1, 1) & \text{otherwise} \end{cases} \tag{6}$$

where $w \in \{0, 1\}$ and $(w_1, w_2) \in \{(0, 0), (0, 1), (1, 0)\}$ is a pair of hidden bits codded in order to insert an average of $\log_2(3)$ bits in a pixel pair. This equation is derived from the pairwise embedding equation of [8]. Additional embedding cases have also been introduced in order to insert one bit, w, in some (but not all) pixels with $d = 0$.

The pairwise PVO approach uses two thresholds to split the blocks into three categories: uniform, slightly noisy and noisy. Uniform blocks are pairwise embedded with equation (6) (very efficient when the pixels have similar values). Slightly noisy blocks are embedded using the IPVO approach of [4] (which is more resilient to value variations in the block) and noisy blocks remain unchanged.

3. PROPOSED SCHEME

The proposed approach is derived from the pairwise PVO scheme of [5] presented in section 2.2. Two complexity thresholds, t_1 and t_2, are used to split the blocks into three groups. The uniform blocks are modified with an improved pairwise PVO approach, the slightly noisy blocks are modified by the scheme described in section 2.1 and the remaining blocks are kept unchanged. A linear programming model is also introduced in order to determine, for a given payload, the optimal values for the thresholds t_1, t_2 and n, the size of the block.

3.1. Pairwise IPVO

The host image is split into non-overlapping equally sized blocks. For each block, the pixels are sorted in ascending order: $x_{\sigma(1)} \leq x_{\sigma(2)} \leq ... \leq x_{\sigma(n)}$. The complexity of the block is determined as:

$$l_c = \frac{4}{n_c} \sum_{i=1}^{n_c} |x_i - \mu_c|, \; x_i \in b_c \tag{7}$$

where b_c contains the graylevel values for $x_{\sigma(3)}, ..., x_{\sigma(n-2)}$ and the pixels that can be gained by expanding the current block by one column left and one line down. n_c is the number of pixels in b_c and μ_c is their average. $|x|$ represents the absolute value of x. The result of the sum is amplified by 4 in order to improve the distribution of l_c values when using integer numbers for the t_1 and t_2 complexity thresholds.

For $l_c \leq t_1$, the locations of both the smallest and the largest three pixels of the block are determined. Equation (1) is used to locate the host candidates. These locations are then compared with the ones of the third smallest/largest pixel within the block as follows:

$$\begin{array}{ll} u_{11} = \min(u_1, \sigma(3)) & u_{21} = \min(u_2, \sigma(n-2)) \\ v_{11} = \max(u_1, \sigma(3)) & v_{22} = \max(u_2, \sigma(n-2)) \\ u_{12} = \min(v_1, \sigma(3)) & u_{22} = \min(v_2, \sigma(n-2)) \\ v_{12} = \max(v_1, \sigma(3)) & v_{22} = \max(v_2, \sigma(n-2)) \end{array} \quad (8)$$

Four difference values are further computed:

$$\begin{array}{ll} d_{l1} = x_{u_{11}} - x_{v_{11}} & d_{r1} = x_{u_{21}} - x_{v_{21}} \\ d_{l2} = x_{u_{12}} - x_{v_{12}} & d_{r2} = x_{u_{22}} - x_{v_{22}} \end{array} \quad (9)$$

Since the above equations produce negative values, the proposed scheme can fully exploit the pixels with $d = 0$ (as in [4]).

The pixels are selected as (x_{u_1}, x_{v_1}) and (x_{u_2}, x_{v_2}) pairs, which are further classified into five groups:

$$(x_{u_1}, x_{v_1}) \in \begin{cases} A & \text{if } d_{l1} \in \{0,1\} \text{ and } d_{l2} \in \{0,1\} \\ B & \text{if } d_{l1} \in \{-1,2\} \text{ and } d_{l2} \in \{-1,2\} \\ C & \text{if } d_{l1} \in \{0,1\} \text{ and } d_{l2} \notin \{0,1\} \\ D & \text{if } d_{l1} \notin \{0,1\} \text{ and } d_{l2} \in \{0,1\} \\ E & \text{otherwise} \end{cases} \quad (10)$$

The pairs are then modified according to their group:

$$(x'_{u_1}, x'_{v_1}) = \begin{cases} (x_{u_1}, x_{v_1}) - (w_1, w_2) & \text{if } (x_{u_1}, x_{v_1}) \in A \\ (x_{u_1}, x_{v_1}) - (w, w) & \text{if } (x_{u_1}, x_{v_1}) \in B \\ (x_{u_1}, x_{v_1}) - (w, 1) & \text{if } (x_{u_1}, x_{v_1}) \in C \\ (x_{u_1}, x_{v_1}) - (1, w) & \text{if } (x_{u_1}, x_{v_1}) \in D \\ (x_{u_1}, x_{v_1}) - (1, 1) & \text{if } (x_{u_1}, x_{v_1}) \in E \end{cases} \quad (11)$$

where $(w_1, w_2) \in \{(0,0), (0,1), (1,0)\}$ and $w \in \{0,1\}$. These equations are equivalent to the ones of [8], but the extra embedding cases from (6) are no longer needed.

The blocks with $t_1 < l_c \leq t_2$ are modified by using the embedding scheme described in Section 2.1. Blocks with $l_c > t_2$ remain unchanged. The overflow/underflow problem is solved using an overflow map that records the location of the corresponding blocks. The overflow map is losslessly compressed and stored in a reserved area of the image by LSB substitution, the original LSBs are stored together with the hidden data. The decoding equations can be easily deduced from their embedding counterparts.

3.2. Linear programming model

In this section a linear programming (LP) model is introduced in order to determine the optimal embedding parameters for a given payload. The parameters are: t_1, t_2 (the two complexity thresholds) and b (an index corresponding to the size of the pixel blocks). Their optimal values are determined by the LP model as pairs: (t_1, b_1) and (t_2, b_2) with $b_1 = b_2$.

Table 1. Embedding parameters for *Lena* and *Mandrill*.

Test image	Capacity [bits]	Embedding parameters			PSNR [dB]	
		t_1	t_2	Block	Estimated	True
Lena	10,000	8	9	4×4	60.881	60.882
	25,000	10	16	3×3	55.767	55.77
	45,000	28	126	3×2	51.317	51.312
Mandrill	10,000	21	42	2×3	54.906	54.908
	14,000	35	126	3×2	51.962	51.963

The LP uses four precomputed matrices. These matrices consider the payload size and distortion for each of the two PVO methods (pairwise IPVO and classic IPVO). The matrices corresponding to the proposed pairwise IPVO approach are first determined.

Let $N_g(t, b)$ be the total number of pixel pairs belonging to group g generated with a block size corresponding to index b and a local complexity value of $l_c \leq t$. The group $g \in \{A, B, C, D, E\}$ is determined with (10) and l_c is computed with (7). For this method, each element of the payload size matrix C_{p1} is computed based on the embedding cases:

$$C_{p1}(t, b) = \log_2 3\, N_A(t, b) + N_B(t, b) + N_C(t, b) + N_D(t, b) \quad (12)$$

The elements of the distortion matrix D_{p1} are also determined:

$$\begin{aligned} D_{p1}(t, b) = 0.67\, N_A(t, b) + N_B(t, b) + 1.5\, N_C(t, b) \\ + 1.5\, N_D(t, b) + 2\, N_E(t, b) \end{aligned} \quad (13)$$

Note that pairs belonging to group A are distorted by (b_1, b_2) (see equation (11)). When $(b_1, b_2) = (0, 0)$ no distortion is introduced. For the remaining two possibilities, a single pixel is modified by one graylevel. Therefore the average distortion for pairs belonging to A is $\frac{1}{3} \cdot 0 + \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 1 = \frac{2}{3} \approx 0.67$. The same approach was used to determine the distortion for the other groups.

The classic IPVO approach is used for the blocks with $t_1 < l_c \leq t_2$, therefore the results depend on both t_1 and t_2. In order to avoid a needless increase in complexity, the two corresponding matrices are computed using a single threshold t (which is later accounted for in order to provide the correct results):

$$\begin{aligned} C_{p2}(t, b) &= H_0(t, b) + H_1(t, b) \\ D_{p2}(t, b) &= 0.5 \cdot H_0(t, b) + 0.5 \cdot H_1(t, b) \\ &\quad + \sum_{i=-255}^{-1} H_i(t, b) + \sum_{i=2}^{255} H_i(t, b) \end{aligned} \quad (14)$$

where $H_i(t, b)$ is the total number of pixels with difference values equal to i (either d_l or d_r from (2)) belonging to b blocks with $l_c \leq t$.

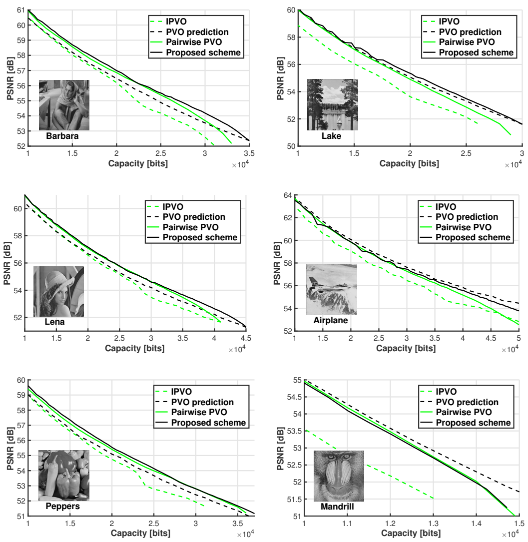

Fig. 1. Capacity/PSNR results for the IPVO scheme of [4], the PVO prediction based scheme of [7], the pairwise PVO scheme of [5] and the proposed pairwise IPVO scheme.

The total payload size/distortion is computed as:

$$\begin{aligned} C_{p\ t_1,b_1,t_2,b_2} &= C_{p1}(t_1,b_1) + C_{p2}(t_2,b_2) - C_{p2}(t_1,b_2) \\ D_{p\ t_1,b_1,t_2,b_2} &= D_{p1}(t_1,b_1) + D_{p2}(t_2,b_2) - D_{p2}(t_1,b_2) \end{aligned} \tag{15}$$

The third term in both equations is needed in order to compensate for the single threshold used in (14). Since the blocks with $l_c < t_1$ are embedded using pairwise IPVO, their contribution to the classic IPVO scheme should be removed.

The proposed LP model is derived from [9]. The problem consists in determining the embedding parameters that provide the desired payload size C_W at minimum distortion. The objective function F is:

$$F = \arg \min_{t_1,b_1,t_2,b_2} D_{p\ t_1,b_1,t_2,b_2} \tag{16}$$

subject to $C_{p\ t_1,b_1,t_2,b_2} \geq C_W$, $t_1 < t_2$ and $b_1 = b_2$.

4. EXPERIMENTAL RESULTS

In this section, we evaluate the effectiveness of the proposed pairwise IPVO reversible data hiding scheme and its LP model. The test are performed using pixel blocks with sizes between 6 and 16 pixels: 2×3, 3×2, 3×3, 3×4, 4×3 and 4×4.

The proposed LP model can be solved using any general LP solver. We have considered the Fico Xpress MP software [10]. A sample of embedding parameters is provided in Table 1. The estimated PSNR is determined based on the distortion model of equation (15) and the true one is obtained by embedding with the corresponding parameters. As can be seen from Table 1, the difference between the estimated distortion and the true one is negligible.

Table 2. Experimental results on Kodak set for the proposed scheme and the pairwise PVO of [5].

Test image	PSNR results for a given capacity [dB]					
	10,000 bits		20,000 bits		30,000 bits	
	Pairwise PVO	Proposed scheme	Pairwise PVO	Proposed scheme	Pairwise PVO	Proposed scheme
Kodim01	63.52	64.07	58	59.42	53.81	55.39
Kodim02	64.26	64.3	60.79	60.76	58.59	58.84
Kodim03	65.55	65.38	61.59	62.43	59.44	60.31
Kodim04	64.09	63.91	59.94	60.42	57.72	58.19
Kodim05	62.27	63.03	58.32	58.77	54.67	55.85
Kodim06	66.09	66.22	61.41	62.64	58.38	60.02
Kodim07	65.11	64.64	61.42	62.13	59.22	60.08
Kodim08	59.87	60.54	55.7	55.89	-	-
Kodim09	63.84	63.44	60.05	60.45	57.82	58.66
Kodim10	63.44	63.34	59.81	60.30	57.51	58.17
Kodim11	65.26	65.73	61.29	61.82	58.23	59.50
Kodim12	64.74	64.64	60.75	61.55	58.38	59.32
Kodim13	58.29	58.19	52.1	52.23	-	-
Kodim14	62.06	62.49	57.83	58.26	54.73	55.57
Kodim15	65.61	65.02	61.45	61.86	59.24	59.8
Kodim16	65.07	65.08	61.01	61.92	59.20	59.76
Kodim17	64.07	64.26	59.83	60.8	57.86	58.31
Kodim18	61.32	61.8	57.32	57.8	54.40	55.06
Kodim19	63.4	63.38	59.89	60.13	57.60	57.86
Kodim20	65.53	62.55	62.47	62.63	60.42	60.73
Kodim21	63.87	63.75	60.04	60.81	57.61	58.5
Kodim22	63.07	63.4	59.22	59.6	56.50	57.05
Kodim23	64.89	64.44	60.87	61.45	58.87	59.64
Kodim24	63.18	62.57	59.99	58.8	56.96	56.02
Average	63.68	63.71	59.63	60.11	57.6	58.3

The experimental results on six classic test images for the proposed scheme are compared with the ones reported in [4], [5] and [7]. The plots are shown in Figure 1. The proposed approach outperforms the other three schemes on four out of the six test images. While [7] has good results on three test images, its performance is limited on the other three. Note that [7] uses the PE-HS/PEE embedding framework.

The performance of the proposed scheme is also evaluated on the Kodak image set. The results are shown in Table 2. As can be seen from Table 2, the proposed pairwise IPVO scheme clearly outperforms the original pairwise PVO scheme of [5].

5. CONCLUSIONS

An original pairwise PVO scheme with improved difference equations and a streamlined pair classification/embedding algorithm has been proposed. Optimized embedding parameters are derived by using a linear programming model. The newly proposed PVO approach appears to outperform state-of-the-art PVO schemes.

6. REFERENCES

[1] Z. Ni, Y. Q.Shi, N. Ansari and W. Su, "Reversible data hiding", *IEEE Transactions on circuits and systems for video technology*, 16 (3), pp. 354-362, 2006.

[2] Y. Q. Shi, X. Li, X. Zhang, H. T. Wu and K. Ma, "Reversible data hiding: advances in the past two decades", *IEEE Access*, 4, pp. 3210-3237, 2017.

[3] X. Li, J. Li, B. Li and B. Yang, "High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion", *Signal Process.*, 93 (1), pp. 198–205, 2013.

[4] F. Peng, X. Li and B. Yang, "Improved PVO-based reversible data hiding", *Digit. Signal Process.*, 25, pp. 255–265, 2014.

[5] B. Ou, X. Li and J. Wang, "High-fidelity reversible data hiding based on pixel-value-ordering and pairwise prediction-error expansion", *Journal of Visual Communication and Image Representation*, 39, pp. 12–23, 2016.

[6] X. Qu and H.J. Kim, "Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding", *Signal Process.*, 111, pp. 249–260, 2015.

[7] B. Ou, X. Li and J. Wang, "Improved PVO-based reversible data hiding: A new implementation based on multiple histograms modification", *Journal of Visual Communication and Image Representation*, 38, pp. 328–339, 2016.

[8] B. Ou, X. Li, Y. Zhao, R. Ni and Y.-Q. Shi, "Pairwise Prediction-Error Expansion for Efficient Reversible Data Hiding", *IEEE Trans. on Image Processing*, 22 (12), pp. 5010–5021, 2013.

[9] I. Caciula and D. Coltuc, "Improved control for low bitrate reversible watermarking", *Proc. IEEE Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1549–1552, 2014.

[10] FICO Xpress MP, *http://www.fico.com/*