Adaptive Pairing Reversible Watermarking

Ioan-Catalin Dragoi, *Member, IEEE*, and Dinu Coltuc, *Senior Member, IEEE*

***Abstract*— This letter revisits the pairwise reversible watermarking scheme of Ou *et al.*, 2013. An adaptive pixel pairing that considers only pixels with similar prediction errors is introduced. This adaptive approach provides an increased number of pixel pairs where both pixels are embedded and decreases the number of shifted pixels. The adaptive pairwise reversible watermarking outperforms the state-of-the-art low embedding bit-rate schemes proposed so far.**

***Index Terms*— Reversible watermarking, pixel pairing, 2D histogram.**

I. Introduction

Reversible watermarking (RW) ensures not only exact extraction of the embedded message, but also exact recovery of the host. For the case of low image embedding bit-rates ($<0.1-0.2$ bpp), a very efficient RW scheme is [2], which embeds data not into the prediction error histogram of single pixels, but into the one of pixel pairs. When both pixels of a pair can be watermarked, instead of embedding two bits of data, i.e. the combinations of bits "00", "01", "10" and "11", the last combination, "11", is eliminated. The embedding of only $\log_2 3$ instead of two bits is compensated by the reduction of the distortion. Until the publication of [3], the pairwise embedding of [2] has been the most efficient low embedding bit-rate RW scheme. The scheme of [3] considers the standard pixel embedding by histogram shifting, but splits the prediction error histogram and optimizes the selection of bins over the multiple sub-histograms.

This letter revisits the pairwise embedding of [2] and, instead of fixed pixel pairing, introduces adaptive pairing. The current pixel is paired with one of its neighbors in order to promote the embedding into both pixels of the pair. A pixel classification procedure ensures both the pairing at embedding and its recovery at detection. The proposed RW scheme not only outperforms the one of [2], but also the one of [3].

II. Proposed Reversible Watermarking Scheme

The proposed scheme is inspired by the pairwise RW of [2]. Data is embedded into the 2D prediction error histogram computed for the rhombus predictor (see [1]) after the usual image splitting and sorting. The main novelties are a context-based pixel classification stage and the adaptive pairing. The major details of the scheme follow.

First, the image is split into two distinct sets forming a chessboard pattern: cross and dot. The prediction context (Fig. 1) for each pixel in the cross set is formed of pixels from the dot set and vice-versa. The processing starts with the cross set.

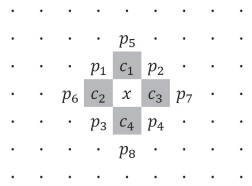

Fig. 1. Prediction context (c_1–c_4) and possible pairing partners (p_1–p_8) for the pixel x.

The local complexity, l_c, of each pixel in the set is computed:

$$l_c = (c_1 - c_2)^2 + (c_2 - c_4)^2 + (c_4 - c_3)^2 + (c_3 - c_1)^2 \tag{1}$$

Based on l_c, the pixels in the set are sorted in ascending order and only the first k pixels are further considered for embedding. The integer k controls the embedding bit-rate.

Each selected pixel x of the current set is predicted as the average of its context (c_1, c_2, c_3 and c_4) and the prediction error e_x is computed. Two error values are selected (R and L) to serve as the primary hosts for hidden data. Based on these values, x is assigned to group A, B or C as follows:

$$x \in \begin{cases} A, & \text{if } L < e_x < R \\ B, & \text{if } e_x > R+1 \text{ or } e_x < L-1 \\ C, & \text{if } e_x \in \{L-1, L, R, R+1\} \end{cases} \tag{2}$$

The k sorted pixels are further processed one by one. The pixels belonging to group A are not used for embedding and remain unchanged. The pixels belonging to B are shifted in order to prevent overlapping with the watermarked pixels from C:

$$x' = \begin{cases} x+1 & \text{if } e_x > R+1 \\ x-1 & \text{if } e_x > L-1 \end{cases} \tag{3}$$

Finally, the pixels of C are adaptively paired and watermarked.

Let x^j, $j = 1, \ldots, n$ ($n \leq k$), be the pixels in the sequence belonging to C. The pairing starts by checking if any of the diagonal neighbors of the current pixel x^j, p_i^j, $i = 1, \ldots, 4$, belongs to C (see Fig. 1). The checking is performed in the increasing order of the index i. Once such a neighbor is found, the current pixel and the neighbor are eliminated from C and the pairing proceeds with the next pixel in the sequence. After the first pass, the pixels without a pairing partner are rechecked for pairing with more distant neighbors: p_5^j, p_6^j, p_7^j and p_8^j. As above, once paired, the pixels are eliminated from C and the next pixel is considered. After the second pass is completed, the remaining pixels belonging to C are paired two by two. This final pairing is performed in the increasing order of the index j, regardless of pixel locations in the image. If n is an odd number, the last pixel of the sequence cannot be paired and it is left unchanged. The neighbors belonging to B checked during pairing are shifted (equation (3)) even if they are not among the k selected pixels.

Manuscript received November 1, 2015; revised February 25, 2016; accepted March 24, 2016. Date of publication March 31, 2016; date of current version April 14, 2016. This work was supported by the Romanian Executive Agency for Higher Education, Research, Development and Innovation under Grant PN-II-PT-PCCA-2013-4-0201, Grant PN-IIPT-PCCA-2011-3.2-1162, and Grant PNII-PT-PCCA-2013-4-1762. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xiaochun Cao.

The authors are with the Faculty of Electrical Engineering, Electronics and Information Technology, Valahia University of Târgoviște, Târgoviște 130024, Romania (e-mail: catalin.dragoi@valahia.ro; dinu.coltuc@valahia.ro).

Digital Object Identifier 10.1109/TIP.2016.2549458

Since neighboring pixels are better correlated than pixels further apart, the proposed scheme prioritizes the pairing of close neighbors and only uses distant partners when no suitable close neighbors are found. The classification in groups allows the proposed scheme to use only pair pixels with similar prediction errors, as opposed to [2] where the pairing is fixed (only (x, p_4) pairs are used, regardless of e_x and e_p).

The two selected prediction errors, R and L, determine four quadrants into the 2D prediction error plane. These quadrants are located at North-East, abbreviated NE, (with the lower left corner in (R, R)), SE (upper left corner in (L, R)), SW (upper right corner in (L, L)) and NW (lower right corner in (R, L)). As in [2], we present the embedding into the NE quadrant. The equations corresponding to the other three quadrants can be easily deduced from the following explanations.

The two pixels of the pair (x, p) are modified (marked or shifted) based on their prediction error. There are four distinct cases: both pixels are marked with $\log_2 3$ bits (mm_1) or 1 bit (mm_2), x is marked and p is shifted (ms) or x is shifted and p is marked (sm). The embedding proceeds as follows:

$$(x', p') = \begin{cases} (x, p) + (b_1, b_2), & \text{if } (e_x, e_p) = (R, R) \quad (mm_1) \\ (x, p) + (b, b), & \text{if } (e_x, e_p) = (R+1, R+1) \quad (mm_2) \\ (x, p) + (b, 1), & \text{if } (e_x, e_p) = (R, R+1) \quad (ms) \\ (x, p_i) + (1, b), & \text{if } (e_x, e_p) = (R+1, R) \quad (sm) \end{cases} \tag{4}$$

where $(b_1, b_2) \in \{(0,0), (0,1), (1,0)\}$ and $b \in \{0, 1\}$.

If a modified value (x' or p') exceeds the image graylevel range (overflow/underflow), the pair is left unchanged and the pixel positions are stored as additional information. This also applies to the pixels shifted with (3).

Once the cross set is embedded, the processing continues with the pixels of the dot set as described by equations (1)-(4). Before going any further, it should be observed that contrary to [2], there are no pairs where both pixels are shifted (ss).

At the decoding stage, the dot set is processed first. The cross set can only be decoded after the dot set is completely restored. Each pixel in a set is assigned to the same group as in the embedding stage using:

$$x' \in \begin{cases} A, & \text{if } L < e_x < R \\ B, & \text{if } e_x > R+2 \text{ or } e_x < L-2 \\ C, & \text{if } e_x \in \{L-2, L-1, L, R, R+1, R+2\} \end{cases} \tag{5}$$

After that, exactly the same approach from the embedding stage is used to determine the pixels shifted with (3) and the pairs marked with (4). This allows hidden data extraction and original graylevel recovery.

A. Adaptive Pairing Example

In this subsection, a simple example of adaptive pairing is provided. The pixels $x_1, x_2, \ldots, x_{12}$ and their prediction errors are shown in Fig. 2 b) and c), respectively. Let us suppose that the sorting in increasing order based on local context (equation (1)) generates the sequence: x_4, x_1, x_2, x_3, x_6, x_{10}, x_7, x_5, x_8, x_{12}, x_{11}, x_9. If the embedding is performed with $R = 0$ and $L = -1$, the sequence is further split according to equation (2) into the subsequences A, B and C, where A is void, $B = \{x_3, x_7, x_5, x_8\}$ and $C = \{x_4, x_1, x_2, x_6, x_{10}, x_{12}, x_{11}, x_9\}$. Only the pixels in C are paired. As discussed above, three stages of pairing are sequentially performed until C is exhausted. In the first pass, the close diagonal

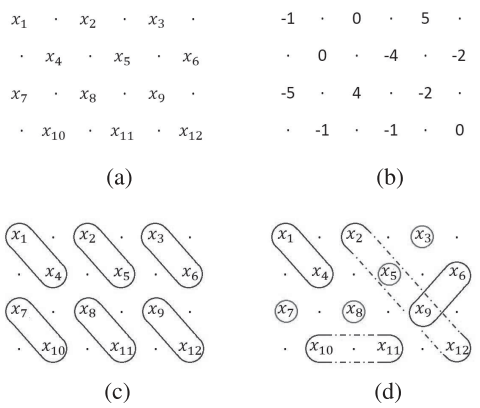

Fig. 2. Pixel pairing at embedding: (a) original pixels; (b) prediction errors; (c) fixed pairing; (d) adaptive pairing.

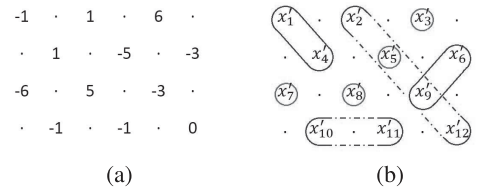

Fig. 3. Pixel pairing at decoding: (a) prediction errors (b) adaptive pairing for the watermarked pixels of Fig. 2.a.

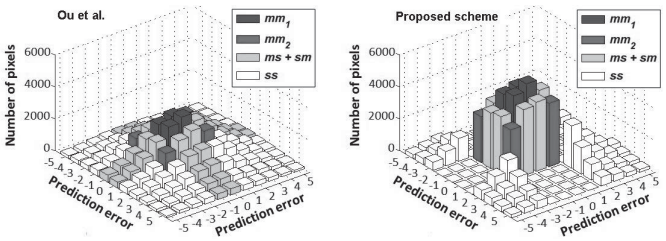

Fig. 4. The 2D histogram used by Ou et al. [2] and the one used by the proposed scheme ($R = 0$, $L = -1$, test image *Lena).*

neighbors are examined, one by one, according to the indexing of Fig. 1. The first pixel in C, x_4, has four diagonal neighbors, x_1, x_2, x_7 and x_8. Since $x_1 \in C$, the search is stopped, the (x_4, x_1) pair is created and both pixels are removed from C. The updated C is $\{x_2, x_6, x_{10}, x_{12}, x_{11}, x_9\}$. The pairing continues with x_2 and its neighbors, x_4 and x_5. Since neither x_4, nor x_5 belong to the updated C, x_2 cannot be paired at this moment and the next pixel, x_6, is considered. Its first diagonal neighbor $x_3 \notin C$. Since its second neighbor $x_9 \in C$, the (x_6, x_9) pair is created and C is updated as $\{x_2, x_{10}, x_{12}, x_{11}\}$. The remaining pixels, x_{10}, x_{12} and x_{11}, do not have valid diagonal partners. The second stage of pairing scans C and checks the distant horizontal and vertical neighbors. While x_2 cannot be paired with its neighbors (namely x_1, x_3, x_8), x_{10} can be paired with the second of its two neighbors: x_{11}. Thus, the pair (x_{10}, x_{11}) is created and C becomes $\{x_2, x_{12}\}$. The last pixel in C, x_{12}, cannot be paired with its neighbors, x_6 or x_{11}. The third stage directly pairs, two by two, the remaining pixels of C. The pair (x_2, x_{12}) exhausts C and the procedure ends. The results for the fixed approach of [2] and the described adaptive pairing are shown in Fig. 2 c) and d), respectively.

The pixels of B are shifted (equation (3)) and the ones of C are either embedded or shifted (equation (4)). Let x' be the modified pixels. Let us briefly investigate the pairing at detection. Since the context is the same as at embedding, the same ordering is induced: x'_4, x'_1, x'_2, x'_3, x'_6, x'_{10}, x'_7, x'_5, x'_8, x'_{12}, x'_{11}, x'_9. Obviously, the prediction

TABLE I

DISTRIBUTION OF PIXEL PAIR TYPES FOR THE FIXED PAIRING APPROACH OF OU et al. [2] AND THE PROPOSED ADAPTIVE PAIRING

Test image	Number of pixel pairs								Number of shifted pixels	
	mm_1		mm_2		$ms+sm$		ss		$(ms+sm+2ss)$	
	Ou et al.	**Proposed**	**Ou et al.**	**Proposed**	**Ou et al.**	**Proposed**	**Ou et al.**	**Proposed**	**Ou et al.**	**Proposed**
Lena	10,158	17,909	6,735	12,881	45,512	30,239	67,390	69,021	180,292	168,281
Boat	3,924	10,359	2,996	8,804	31,809	19,082	91,066	91,805	213,941	202,692
Mandrill	1,365	5,461	1,295	5,244	18,879	10,688	108,256	108,657	235,391	228,002
Lake	4,028	9,796	2,790	8,087	29,032	17,573	93,945	94,594	216,922	206,761
Elaine	3,748	9,642	2,379	7,639	28,244	16,510	95,424	96,259	219,092	209,028
Jetplane	23,681	30,091	8,074	12,492	48,113	35,626	49,927	51,840	147,967	139,306

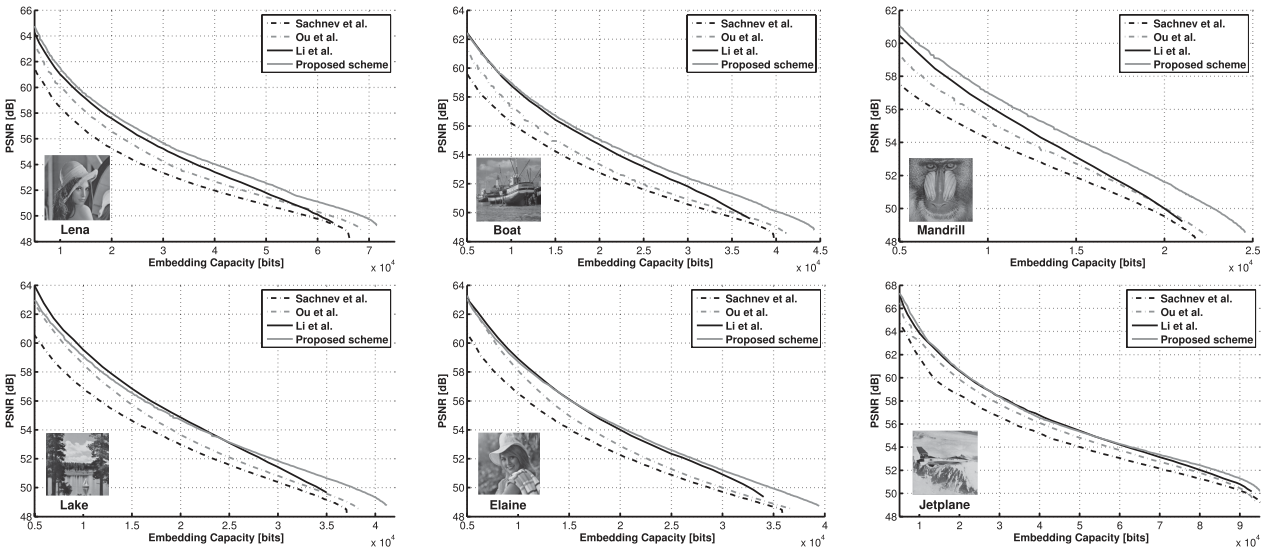

Fig. 5. Capacity/PSNR comparison between the proposed scheme and the ones of Sachnev et al. [1], Ou et al. [2] and Li et al. [3].

errors at detection are different. Let us suppose that at detection one has the prediction errors of Fig. 3.a. One can easily check that for $R = 0$ and $L = -1$, with equation (5) one identically recovers the sequences A, B and $C = \{x'_4, x'_1, x'_2, x'_6, x'_{10}, x'_{12}, x'_{11}, x'_9\}$. As above, the pixels in C are processed one by one starting with x'_4 and the corresponding diagonal neighbors are evaluated as potential pairing partners. Two valid pairs are found (x'_4, x'_1) and (x'_6, x'_9). The group is updated: $C = \{x'_2, x'_{10}, x'_{11}, x'_{12}\}$. Next, for the remaining pixels, the more distant horizontal and vertical neighbors are evaluated as partners. The (x'_{10}, x'_{11}) pair is found and C becomes $\{x'_2, x'_{12}\}$. The remaining pixels in C are directly paired, forming (x'_2, x'_{12}). The same pixel pairs were obtained for both embedding (Fig. 2.d) and decoding (Fig. 3.b).

III. EXPERIMENTAL RESULTS

In this section, experimental results for the proposed adaptive pairing RW scheme are presented. Fig. 4 shows the 2D prediction error histogram of [2] and the one used by the proposed scheme, both for the test image *Lena*. In order to compare the two approaches more easily, the pixels in group B were paired to form shifted pairs (ss).

As is immediately apparent, the 2D histogram produced by the proposed scheme is considerably sharper than the one of [2]. Furthermore, the embedded pairs are completely concentrated in the center of the histogram. The 2D watermarking of the ms, ms and ss pairs produces equivalent results to those of classical RW schemes like [1]. The superior performance of the 2D histogram approaches are caused by the mm_1 and mm_2 pairs. The large number of ms and sm pairs created by the approach of [2] limits its RW performance. As shown in Table I, the adaptive pairing of only pixels that have the potential to be embedded (group C) produces a larger number of mm_1 and mm_2 pairs.

Next, the proposed scheme is compared with the classic approach of [1], the fixed pairing of [2] and the recent scheme of [3] on six 512×512 images extensively used in reversible watermarking. The results are presented in Fig. 5. The proposed scheme outperforms the other approaches on all test images, with an average gain in PSNR of 2.08 dB over [1], 1.15 dB over [2] and 0.32 dB over [3]. On three of the test images (*Lake*, *Elaine* and *Jetplane*) the gain in PSNR of the proposed scheme relative to [3] is rather low: for around two thirds of the embedding capacity domain the two approaches obtain equivalent results, for the remaining capacities the proposed scheme has superior results (a gain of up to 0.6 dB on *Jetplane* and over 1 dB on the other two images).

IV. CONCLUSIONS

An adaptive pairing for the pairwise embedding reversible watermarking scheme of [2] has been proposed. Pixels are paired with one of their neighbors in order to ensure the embedding into both pixels of the pair. A pixel classification procedure ensures both the pairing at embedding and its recovery at detection. The newly proposed scheme outperforms the other low-embedding bit-rate approaches proposed so far.

REFERENCES

[1] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 989–999, Jul. 2009.

[2] B. Ou, X. Li, Y. Zhao, R. Ni, and Y. Q. Shi, "Pairwise prediction-error expansion for efficient reversible data hiding," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5010–5021, Dec. 2013.

[3] X. Li, W. Zhang, X. Gui, and B. Yang, "Efficient reversible data hiding based on multiple histograms modification," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 2016–2027, Sep. 2015.