Improved Pairwise Embedding for High-Fidelity Reversible Data Hiding

Ioan Catalin Dragoi
Electrical Engineering Dept.
Valahia University of Targoviste, Romania
Email: catalin.dragoi@valahia.ro

Dinu Coltuc
Electrical Engineering Dept.
Valahia University of Targoviste, Romania
Email: dinu.coltuc@valahia.ro

***Abstract*—Pairwise reversible data hiding (RDH) restricts the embedding to 3 combinations of bits per pixel pair ("00", "01", "10"), by eliminating the embedding of "1" into both pixels. The gain in quality is significant and the loss in embedding bit-rate is compensated by embedding into previously shifted pairs. This restriction requires a special coding procedure to format the encrypted hidden data. This paper proposes a new set of embedding equations for pairwise RDH. The proposed approach inserts either one or two data bits into each pair based on its type, bypassing the need for special coding. The proposed equations can be easily integrated in most pairwise reversible data hiding frameworks. They also provide more room for data embedding than their classic counterparts at the low embedding distortion required for high-fidelity RDH.**

I. Introduction

Reversible data hiding (RDH) allows for both the extraction of the hidden data and the exact recovery of the host signal. Image RDH exploits the correlation between neighboring pixels in order to efficiently insert the hidden data into the host image. High-fidelity RDH has a severe limitation on the distortion caused by the embedding algorithm, usually requiring for the host pixel to be modified at most by one graylevel value.

Most RDH algorithms embed data into individual pixel using the prediction error histogram shifting framework (PE-HS) of [1] or the prediction error expansion framework (PEE) of [2]. Even though they were developed separately, PEE can be viewed as a generalized form of PE-HS. PEE was also further refined in [3] with the introduction of local complexity based capacity control and non-causal prediction.

Recently, two different approaches were developed for high-fidelity RDH, namely block based pixel-value-ordering (introduced in [4]) and pairwise embedding (introduced in [5]). The latter outperforms both PE-HS/PEE and pixel-value-ordering.

Pairwise RDH schemes process the image pixels as pairs. Based on their corresponding prediction errors, the host pairs will either contain an average of $\log_2 3$ bits or one bit. Pairs that cannot be used as hosts have their values shifted by one graylevel in order to maintain reversibility. [5] formed the pairs by grouping two diagonally connected pixels and used their horizontal and vertical neighbors as the prediction context. [6] modified the pairwise embedding process in order to allow the pairing of horizontally connected pixels. A pixel triplet RDH framework derived from [5] was introduced in [7]. Pairwise embedding was also adapted for the block based pixel-value-ordering framework in [8] by processing the four host pixels from each block as two sets of pairs. The basic pairwise framework was significantly improved with the introduction of adaptive pairing in [9]. The prediction error is used to distribute the image pixels into groups. Only the pixels that are best suited for pairwise embedding are paired with each-other. The pixels that do not fulfill the pairing requirements are individually shifted. The pairing algorithm of [9] produces better quality pairs, which in turn significantly improve the efficiency of pairwise RDH. [10] uses geodesic paths as a pairing criteria for a block based form of adaptive pairing. A modified version of [9] that allows for the horizontally connected pixels to be adaptively paired was also proposed in [11].

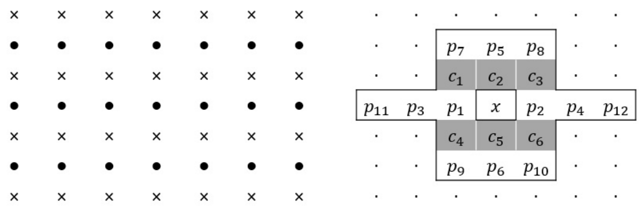
In order to embed an average of $\log_2 3$ bits in certain pairs and one bit in others, all pairwise RDH schemes require a three symbol based coding of part of the encrypted hidden data. This is a cumbersome procedure that depends both on the embedding parameters and on the prediction error distribution of the pairs. This paper introduces a new set of embedding equations for pairwise RDH. Thus, the pairs that contained $\log_2 3$ bits are now embedded with either one or two bits. This approach completely removes the need for the extra coding, without compromising the efficiency of pairwise RDH. On the contrary, the proposed equations provide higher embedding capacities at lower distortion.

The outline of the paper is as follows. The pairwise RDH framework introduced in [11] is briefly described in Section II. The proposed embedding equations are presented and discussed in Section III. The experimental results are presented in Sections IV. The conclusions are then drawn in Section V.

II. Related work

The proposed embedding equations can be easily integrated in most pairwise RDH frameworks (including those proposed in [5], [6], [9], [10] and [11]). We shall further describe in this section one of these frameworks, namely the one recently introduced in [11]. There is also a framework that is not compatible with the proposed embedding equations, namely the pairwise pixel-value-ordering introduced in [8]. The proposed approach requires flexibility when it comes to shifting directions. The shifting directions used in [8] cannot

ISBN 978-90-827970-1-5 © EURASIP 2018



(a) (b)

Fig. 1. The cross and dot sets (a), the c_1–c_6 prediction context and the p_1–p_{12} pairing partners for x (b) using the pairing scheme of [11].

be altered. Also note that in terms of performance, [8] is outperformed by [9], [10] and [11].

The pairwise reversible data hiding framework introduced in [11] is derived from the adaptive pairing scheme of [9]. The embedding algorithm first splits the image rows into two disjoint sets, cross and dot, as shown in Fig. 1.a. The first and last lines and columns of the image are not directly used for data hiding. Also note that the other pairwise embedding schemes, except [6] and [8], use the chessboard pattern introduced in RDH by [3].

The cross set is processed first. For each pixel in the set, a local complexity value is computed based on the c_1–c_6 prediction context (see Fig. 1.b):

$$\begin{aligned} l_c &= (c_1-c_2)^2+(c_2-c_3)^2+(c_3-c_6)^2+(c_1-c_4)^2 \\ &+(c_4-c_5)^2+(c_5-c_6)^2 \end{aligned} \tag{1}$$

The pixels in the current set are then sorted in ascending order based on their l_c value. Only the first N sorted pixels are further used for data hiding (the remaining pixels are not altered). Each of the N pixels is predicted as:

$$\hat{x} = \left\lfloor \frac{c_1+2c_2+c_3+c_4+2c_5+c_6}{8} + \frac{1}{2} \right\rfloor \tag{2}$$

where $\lfloor a \rfloor$ is the greatest integer less than or equal to a.

The prediction error is then computed:

$$e_x = x - \hat{x} \tag{3}$$

The pixels are distributed into three distinct groups:

$$x \in \begin{cases} A, & \text{if } l < e_x < r \\ B, & \text{if } \; e_x < l-1 \text{ or } e_x > r+1 \\ C, & \text{if } e_x \in \{l-1, l, r, r+1\} \end{cases} \tag{4}$$

where r and l are the two primary prediction error histogram bins selected for data hiding. In the 2D histogram, r and l generate four bins. In order to simplify the presentation, we shall further consider $r = 0$ and $l = -1$. Thus, (4) becomes:

$$x \in \begin{cases} B, & \text{if } e_x < -2 \text{ or } e_x > 1 \\ C, & \text{if } e_x \in \{-2,-1,0,1\} \end{cases} \tag{5}$$

Note that e_x has integer values. There are no integer values between -1 and 0, therefore group A is void. Anyway, the pixels belonging to A are not used by the embedding scheme and remain unaltered).

The pixels belonging to B have their graylevel shifted by one position (to prevent them from overlapping with the marked values from C):

$$x' = \begin{cases} x+1 & \text{if } e_x > 1 \\ x-1 & \text{if } e_x < -2 \end{cases} \tag{6}$$

The pixels in C are then paired with each-other. The pairing process has two stages. In the first stage, the p_1–p_{12} pixels (shown in Fig. 1.b) are considered one-by-one as possible pairing partners for x. These pixels must belong to the same set as x and are positioned around x in order to exploit the correlation between neighboring pixels. If $p_1 \in C$, then the (x, p_1) pair is formed and both pixels are removed from C. They can no longer be selected as pairing partners for other pixels. If $p_1 \notin C$, then p_2 can be evaluated by checking if $p_2 \in C$. The procedure is repeated for the remaining pairing candidates, until a pairing partner is found or all twelve candidates are checked. The second pairing stage is performed on the remaining pixels in C after the first stage concluded. In this stage, x is paired with the next pixel belonging to C in the local complexity sorting order.

Each (x, p) pixel pair is then embedded with hidden data:

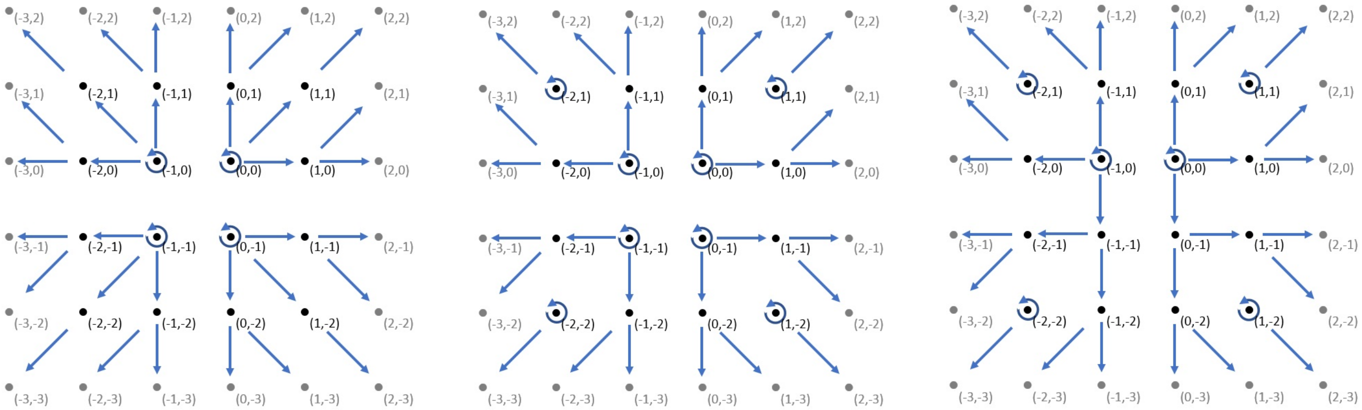
$$(x', p') = \begin{cases} (x,p)+(s_x b_1, s_p b_2), & \text{if } e_x \in \{-1,0\} \text{ and } e_p \in \{-1,0\} \\ (x,p)+(s_x b, s_p b), & \text{if } e_x \in \{-2,1\} \text{ and } e_p \in \{-2,1\} \\ (x,p)+(s_x b, s_p), & \text{if } e_x \in \{-1,0\} \text{ and } e_p \in \{-2,1\} \\ (x,p)+(s_x, s_p b), & \text{if } e_x \in \{-2,1\} \text{ and } e_p \in \{-1,0\} \end{cases} \tag{7}$$

where $s_i = \begin{cases} 1, & \text{if } e_i \geq 0 \\ -1, & \text{if } e_i \leq -1 \end{cases}$ is the shifting direction. The $b_i \in \{0, 1\}$ encrypted hidden data bits are either directly embedded or are first coded using a three symbol based alphabet. These symbols are embedded as $(b_1, b_2) \in \{(0,0), (0,1), (1,0)\}$. In other words, a pixel pair will either contain one bit of hidden data (when the bit is directly embedded) or an average of $\log_2 3$ bits (when using codded bits). This codding is a cumbersome aspect of pairwise embedding that is found in all pairwise data hiding frameworks. Before the embedding process can start, the data hider must determine the appropriate distribution of coded/not-coded bits using the current pixel set 2D prediction error histogram. Note that one can use a simple (suboptimal) coding approach for (b_1, b_2): embed two bits when $b_1 = 0$, otherwise $(b_1, b_2) = (1, 0)$ and a single bit, b_1, is embedded. This suboptimal coding decreases the capacity for such pairs from $\log_2 3 \approx 1.57$ bits to 1.5 bits.

The 2D prediction error histogram mapping of equation (7) is shown in Fig. 2.b together with the ones of classic PEHS/PEE and of the proposed pairwise embedding (discussed in the next section).

The entire embedding process is then repeated for the dot set: the pixels are sorted based on (1) using a prediction context with modified values from the cross set; the first N sorted pixels are selected; the prediction error is determined for each selected pixel with (2) based on (3); the N pixels are first grouped with (5) and then shifted with (6) or marked with (7). The locations of the last embedded pixels/pairs for each set are recorded and stored together with r, l, N and

ISBN 978-90-827970-1-5 © EURASIP 2018



(a) Classic PE-HS/PEE (b) Pairwise embedding (c) Proposed embedding

Fig. 2. The effect of the embedding equations on the (e_x, e_p) prediction error pairs.

an overflow map by LSB substitution in a reserved area. The original reserved area LSBs are appended to the hidden data before the embedding stage.

The decoding stage starts by reading the auxiliary data from the reserved area LSBs. The dot set is processed first. The decoding process is similar to the embedding one: the pixels in the current set are sorted based on their corresponding local complexity value, the prediction errors are determined and the pixels are grouped using the following grouping selection:

$$x' \in \begin{cases} B, & \text{if } e_x < -3 \text{ or } e_x > 2 \\ C, & \text{if } e_x \in \{-3,-2,-1,0,1,2\} \end{cases} \tag{8}$$

Note that (8) produces exactly the same groups as (5). The bounds of the groups in (8) are updated to compensate for the distortion introduced by the embedding algorithm.

The pixels in B are restored by reversing equation (6). The pixels in C are paired using the previously described two staged pairing process (the pixels are processed in the same order). Their values are restored by reverting (7) (the reverse mapping operation from Fig. 2.b). The same decoding process is then repeated for the cross set. The original reserved area LSBs are also restored.

III. Proposed Embedding Scheme

The goal of the proposed embedding scheme is to remove the coding associated with pairwise embedding without compromising its performance. The proposed scheme only alters the embedding equations for the four main embedding pairs, namely those with $(e_x, e_p) \in \{(0,0),(-1,0),(0,-1),(-1,-1)\}$ (corresponding to the first embedding case in equation (7)). The proposed embedding equations for these pairs are as follows:

- If $(e_x, e_p) \in \{(0,0),(-1,0)\}$, then two hidden bits are inserted into the (x,p) pixel pair:

$$(x',p') = \begin{cases} (x,p), & \text{if } (b_1,b_2) = (0,0) \\ (x,p) + (0,s_p), & \text{if } (b_1,b_2) = (0,1) \\ (x,p) + (s_x,0), & \text{if } (b_1,b_2) = (1,0) \\ (x,p) + (0,-s_p), & \text{if } (b_1,b_2) = (1,1) \end{cases} \tag{9}$$

where $s_i = \begin{cases} 1, & \text{if } e_i \geq 0 \\ -1, & \text{if } e_i \leq -1 \end{cases}$ is the previously discussed shifting direction.

- If $(e_x, e_p) \in \{(0,-1),(-1,-1)\}$, then one hidden bit is inserted into (x,p):

$$(x',p') = \begin{cases} (x,p) + (0,s_p), & \text{if } b = 0 \\ (x,p) + (s_x,0), & \text{if } b = 1 \end{cases} \tag{10}$$

The remaining pairs are processed using standard pairwise embedding:

$$(x',p') = \begin{cases} (x,p) + (s_x b, s_p b), & \text{if } e_x \in \{-2,1\} \text{ and } e_p \in \{-2,1\} \\ (x,p) + (s_x b, s_p), & \text{if } e_x \in \{-1,0\} \text{ and } e_p \in \{-2,1\} \\ (x,p) + (s_x, s_p b), & \text{if } e_x \in \{-2,1\} \text{ and } e_p \in \{-1,0\} \end{cases} \tag{11}$$

The effect of the proposed embedding equations on the prediction error is shown in Fig. 2.c.

Note that the proposed scheme is conceptually similar to the suboptimal coding described in the previous section. For both approaches the main embedding pairs contain either one, or two bits of data. The key difference is in the way the pairs are selected. The suboptimal coding selects them based on the encrypted hidden data, i.e., essentially at random. The proposed scheme embeds two bits in the pairs with $(e_x, e_p) \in \{(0,0),(-1,0)\}$. But $e_x = 0$ is the most common prediction error, therefore $(0,0)$ is the most common error pair. In other words, the $(0,0)$ bin of the 2D prediction error histogram has the largest number of pairs. $(-1,0)$ and $(-1,0)$ represent the second and third most populated histogram bins (with interchangeable positions based on the host image). $(-1,-1)$ is usually the fourth most populated bin. Therefore, the proposed embedding scheme inserts two bits into each pixel pair of the two most populated histogram bins at the cost of embedding one bit in pairs belonging to the third and fourth most populated bins. This aspect allows the proposed approach to outperform both the standard three symbol coding and the suboptimal coding.

One can notice from Fig. 2.b that for standard pairwise embedding, the four embedding quadrants can move independently of each-other, therefore one can control the provided

ISBN 978-90-827970-1-5 © EURASIP 2018

Fig. 3. The test images: *Lena*, *Mandrill*, *Jetplane*, *Barbara*, *Elaine*, *Lake*, *Boat* and *Peppers*.

capacity by changing the primary embedding bins (r and l). Note that the capacity is still primarily controlled with N. In Fig. 2.c, the quadrants are vertically connected. Therefore, the connected quadrants cannot move independently from each-other. Nevertheless, in terms of performance, this limited bin selection provides a similar level of capacity control as the one offered by classic pairwise embedding.

IV. Experimental results

In this section, the performance of the proposed pairwise embedding equations is evaluated. Eight classic 512×512 graylevel test images intensely used in image processing form the test set. These images are shown in Fig. 3, seven of them (all images except *Barbara*) are available at `http://sipi.usc.edu/database/`. The peak signal-to-noise ratio between the original image and its watermarked version is used to evaluate the distortion introduced by the embedding algorithm (a distortion that is completely removed by the decoding algorithm).

The proposed pairwise embedding equations are compared with their classic counterparts and the results are presented in Fig. 4. A gain in embedding capacity over the classic equations is observed on all test images. This capacity is under the constraint of high-fidelity reversible data hiding: the pixel values are distorted at most by one graylevel value. The proposed equation with the pairwise framework of [11] provides the highest embedding capacities reported in literature under this constraint. A considerable gain in PSNR over the classic approach is obtained on *Mandrill*, but this gain is mainly due to the shape of the 2D prediction error histogram on *Mandrill*, which favors the proposed equations. Good results are also obtained on *Elaine* and *Lake*. For the remaining images, a noticeable gain in PSNR is observed at the higher end of the capacity domain. Similar results were obtained when using the frameworks of [5] and [9].

V. Conclusion

A new set of embedding equations was introduced for pairwise reversible data hiding. These equations can be easily integrated in most pairwise reversible data hiding frameworks. The host pixel pairs contain either one or two hidden data bits, based on their corresponding prediction errors. The pixel pairs do not require a separate coding stage for the hidden data. Furthermore, the proposed equations also provide more embedding room under the constraint of high-fidelity reversible data hiding. The experimental results show the effectiveness of the proposed approach.

Acknowledgment

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

References

[1] W. Hong, T.-S. Chen and C.-W. Shiu. "Reversible data hiding for high quality images using modification of prediction errors," *Journal of Systems and Software.*, vol. 82, no. 11, pp. 18331842, 2009.

[2] D.M. Thodi and J.J. Rodriguez. "Expansion embedding techniques for reversible watermarking," *IEEE Trans. on Image Processing*, vol. 16, no. 3, pp. 721730, 2007.

[3] 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, pp. 989–999, 2009.

[4] 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.

[5] 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*, vol. 22, no. 12, pp. 5010–5021, 2013.

[6] I.-C. Dragoi, D. Coltuc and I. Caciula "Horizontal pairwise reversible watermarking," *Proc. 23rd Eur. Signal Process*, pp. 56–60, 2015.

[7] S. Cai, X. Li, J. Liu and Z. Guo, "A new reversible data hiding scheme exploiting high-dimensional prediction-error histogram," *IEEE Intl. Conf. on Image Processing (ICIP)*, pp. 2732–2736, 2016.

[8] B. Ou, X. Li and J. Wang, "High-fidelity reversible data hiding based on pixel-value-ordering and pairwise prediction-error expansion," *J. Vis. Commun. Image R.*, vol. 39, 12–23, 2016.

[9] I.-C. Dragoi and D. Coltuc, "Adaptive pairing reversible watermarking," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2420–2422, 2016.

[10] B. Ou, X. Li, J. Wang and F. Peng, "High-fidelity reversible data hiding based on geodesic path and pairwise prediction-error expansion," *Neurocomputing*, vol. 226, pp. 23–34, 2017.

[11] I.-C. Dragoi, D. Coltuc and H.G. Coanda, "Adaptive pairwise reversible watermarking with horizontal grouping," *International Symposium on Signals, Circuits and Systems (ISSCS)*, pp. 1–4, 2017.

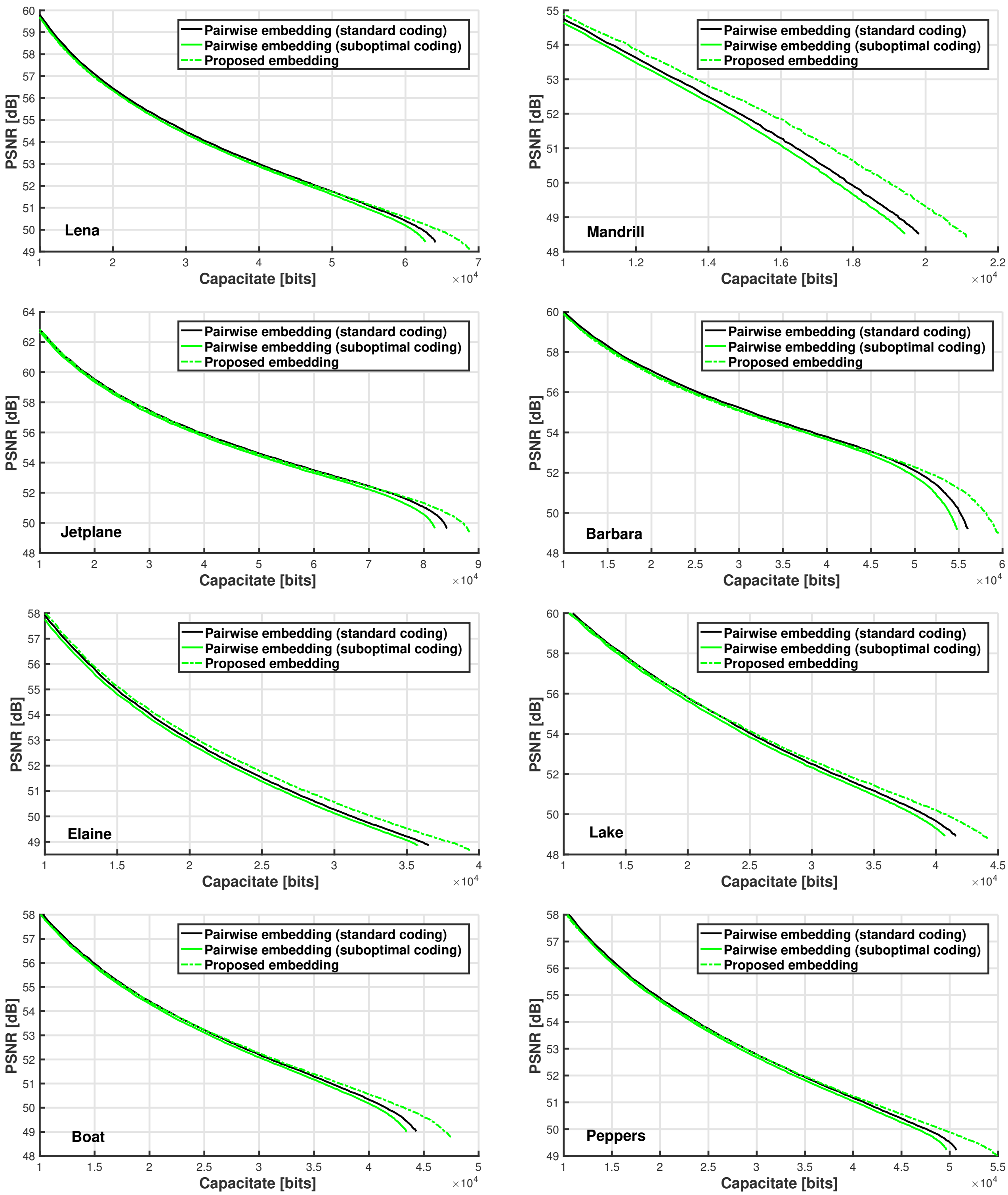


Fig. 4. PSNR/capacity results for the standard and proposed embedding equations, both using the pairwise framework introduced in [11].