Acta Cybernetica **24** (2020) 663–678.

Pixel Grouping of Digital Images for Reversible Data Hiding

Sultan A Hasib[a b] and Hussain Nyeem[a c]

Abstract

Pixel Grouping (PG) of digital images has been a critical consideration in the recent development of the Reversible Data Hiding (RDH) schemes. While a PG kernel can define pixel-groups with the different neighborhoods for better embedding rate-distortion performance, only the group of horizontal neighborhood pixels of size 1×3 has so far been considered. In this paper, we, therefore, construct the PG kernels of sizes 3×1, 2×3 and 3×2, and investigate their potentials to improve both the embedding capacity and the embedded image quality for a PG-based RDH scheme. A kernel of size 3×2 (or 2×3) that creates a pair of pixel-triplets (*i.e.*, two L-shaped blocks) and offers a higher possible correlation among the pixels. These kernels thus can be better utilized for improving a PG-based RDH scheme. Considering this, we develop and present an improved PG-based RDH scheme and the computational model of its key processes. Experimental results demonstrated that our proposed RDH scheme offers reasonably better embedding rate-distortion performance than the original scheme.

Keywords: pixel value ordering, reversible embedding, data hiding, prediction and sorting

1 Introduction

Multimedia data have recently witnessed a tremendous growth that continues with a broader impact on today's life-hood, society, research, and industry. Their uses have shown great promises for the spectrum of emerging applications like different distant and cooperative systems and services in the areas of medical, space, military, security, and surveillance. However, with the advances in communication technologies, their exchange over the public communication network is also raising many security concerns, including forgery, copyright violation, and privacy invasion of multimedia data [6]. To addressing these problems, Reversible Data Hiding (RDH) is being widely investigated [15, 24].

[a]Department of Electrical, Electronic and Communication Engineering (EECE), Military Institute of Science and Technology (MIST), Mirpur Cantonment, Dhaka-1216, Bangladesh

[b]E-mail: hasib_3635@hotmail.com, ORCID: https://orcid.org/0000-0003-1335-0053

[c]E-mail: h.nyeem@eece.mist.ac.bd, ORCID: https://orcid.org/0000-0003-4839-5059

DOI: 10.14232/actacyb.277104

RDH is an evolving forensic and covert-communication technology for multimedia data like digital images. An RDH scheme embeds data into a *cover* image, and the embedded data later can be extracted on-demand basis. An RDH scheme thus has two main processes: *generation* and *embedding* [14, 16]. In the *generation*, the data to be embedded in the cover image are generated and processed as per the requirements of an intended application. The *embedding*, on the other hand, deals with how and where the data are to be embedded in the cover image. The generation process thus deals with the required security properties like integrity and confidentiality, and an embedding technique controls the embedding performance of the RDH scheme.

The embedding rate-distortion criteria mainly determine the embedding performance of an RDH scheme. The *embedding rate* or *embedding capacity* measures how much data can be embedded in a cover image, and the *distortion* measures how much visual quality of the cover is compromised for embedding. Much attention in the data hiding research thus can reasonably be tracked in the development of various embedding techniques with better embedding rate-distortion performance in the last two decades [1–5, 9–13, 17–23, 25, 27].

Among different types of RDH schemes, Pixel Grouping (PG), also called Pixel Value Ordering (PVO), has shown great promises for better embedding rate-distortion performance [10, 12, 19–22] (see Sec. 2). The PG-based schemes thus have the potential to offer a higher embedding rate and lower embedding distortion (*i.e.*, better-embedded image quality). In such schemes, while pixel values are grouped and arranged in a numerical order to better utilize their correlations for improving the embedded image quality, not much attention has been paid in the computation of PG with better pixel correlation.

In this paper, we report an improved PG-based RDH scheme with better utilization of pixels' correlation in pixel grouping. We call each pixel-group an *image-block* in this paper. As will be discussed in Sec. 2, Jung's scheme [10] showed the best possible embedding rate-distortion performance so far in a minimum image-block scenario. We have investigated the case of that scheme [10] that employed image-block of size 1×3 and analyzed the embedding rate-distortion performance of our proposed improvement with other possible block sizes to have better pixel correlation. Notably, in a mixed (*i.e.*, combination of horizontal, vertical, and diagonal) neighborhood, pixels in an image-block remain relatively more correlated. We, therefore, construct and analyze different image-blocks in modeling a PG-based RDH scheme. Thereby, a greater possible pixels' correlation in an image-block can be, utilized in embedding for a better rate-distortion performance.

The remainder of this paper is structured as follows. The current state of the PG-based RDH schemes is reviewed in Sec. 2. We develop and present a general computational model of a PG-based RDH scheme to construct different image-blocks and to examine their effect on the embedding performance in Sec. 3 and analyze the experimental results in Sec. 4. Conclusions are given in Sec. 5.

2 State of RDH schemes

Development of reversible embedding techniques has underpinned different RDH schemes; for example, Difference Expansion (DE) schemes [1,9], Histogram Shifting (HS) schemes [13, 23], Reversible Contrast Matching (RCM) schemes [2, 5], and Prediction Error Expansion (PEE) schemes [3, 4, 10–12, 19–22, 25]. Among them, PEE-based embedding combined the potential of HS and DE techniques to utilize the image redundancy better. Embedding distortion in PEE highly depends on the prediction-error histogram, where a sharp distribution of the histogram offers lower embedding distortion. A better *predictor* is thus always desirable in PEE to obtain a sharper histogram [4].

Additionally, the sorting of prediction errors has been another consideration for improving the performance of PEE-based embedding [23]. Of the sorted prediction errors, the lower values are used for embedding to minimize distortion in the embedded image. Li *et al.* [11] reported that a higher embedding rate with lower distortion is obtainable by embedding in the prediction-errors with lower complexities. Coatrieux *et al.* [3] proposed an adaptive embedding technique that determines the most suitable carrier-class according to its local specificity for data embedding. For better embedding rate-distortion performance, a PEE-based RDH scheme, therefore, aims to utilize correlations of the pixels in an image-block.

The PG technique has lately better utilized the image correlations in PEE-based embedding. Unlike the classical PEE, the PG-based PEE predicts a pixel that has a higher correlation to the original pixels in an image-block. Li *et al.* [12] introduced the PG-based RDH scheme that either increases (or decreases) or keeps unchanged the maximum (or minimum) pixel in a block for embedding 1-bit data. That scheme was later improved with the consideration of dual maximum (or minimum) pixels for prediction errors [21], adaptive prediction of maximum (or minimum) valued pixel [20], pixel-wise PVO [22] and 2D-PVO with pairs of prediction errors [19].

Recently, Jung [10] proposed a scheme that operates on the image-blocks of three pixels, where two successive blocks do not share any pixel. For embedding in each block, its pixel-values are sorted in ascending order to compute the maximum and minimum prediction errors from the maximum and minimum pixels in the block, respectively. That scheme offers better-embedded image quality with reasonably higher embedding capacity.

However, like the other aforementioned PG-based RDH schemes, computation of image-blocks with higher pixels correlation has not been considered. The better utilization of pixels correlation may lead to further improvement of the scheme with better rate-distortion performance. This consideration leads us to investigate different structures of image-blocks for PG-based embedding. Our preliminary results were presented in the conference proceedings [7, 8] that have been extended in this paper with the substantial revision of the model, analysis with more details, and new results.

3 An improved PG-based RDH scheme

In this section, we develop and present a computational model of the PG-based RDH scheme that generally captures the principle of both Jung's scheme [10] and our proposed modifications. The improved PG process is briefly introduced below, followed by our generalized embedding and extraction processes.

3.1 Proposed pixel grouping

A PG-based embedding utilizes image correlations to improve the embedding rate-distortion performance, as mentioned in Sec. 1. The embedding of the Jung's scheme computes the unit prediction error in an image-block of size 1×3, which restricts the block-pixels' correlations to only the horizontal context. Thus, redefining an image-block with both the horizontal and vertical contexts may further improve the embedding rate-distortion performance.

We thus have investigated the embedding performance of the PG-based RDH scheme for different structures of the image-blocks. Unlike the image-blocks used in the Jung's RDH scheme [10], we have employed the other possible structures of an image-block to determine the improvements in the embedding performance. The image-blocks of size 3×1 for vertical orientation and the blocks of size 2×3 and 3×2 for the mixed (*e.g.*, horizontal, vertical, and diagonal) orientations are considered. The image-blocks of both the sizes, 3×2 and 2×3 give a pair of pixel-triplets (*i.e.*, two L-shaped blocks). The construction of two L-shaped blocks illustrated in Fig. 1(c–f) (with the green and blue colors) from a block of 6-pixels means that each L-shaped block is of a fixed size of 3 pixels. These blocks of 3 pixels can be used as the other blocks of 3 pixels like in Fig. 1(a and b) for embedding. Note that Jung [10] used the structure in Fig. 1(a), and the others in Fig. 1(b–f) are studied for the proposed PG-based embedding.

Construction of the structures of an image-block shown in Fig. 1 can be abstracted with the $block\,(\cdot)$ and $de\_block\,(\cdot)$ for the generalized PG-based embedding with an additional input argument σ (see Sec. 3.2). This means, for Jung's scheme, $\sigma = [1, 3]$ defines a block of size 1×3, and for the proposed embedding, $\sigma = [3, 1], [3, 2]$ and $[2, 3]$ define an image-block of size 3×1, 3×2 and 2×3, respectively. With a suitable σ, a PG-based embedding would have more correlated pixels in an image-block to offer better rate-distortion performance.

3.2 PG-based embedding

Let an image, I of size $M \times N$ is to be given as input (or *cover*) image and used for the embedding of secret-data D. The embedding process follows the following steps to output the embedded image I'. As in Algorithm 1, steps of the embedding are discussed below.

Step 1: A set of image-blocks, B is first obtained from an input image, I for a given block-size σ such that $B = \{B_n\}$, where B_n is a set three pixels of the n-th

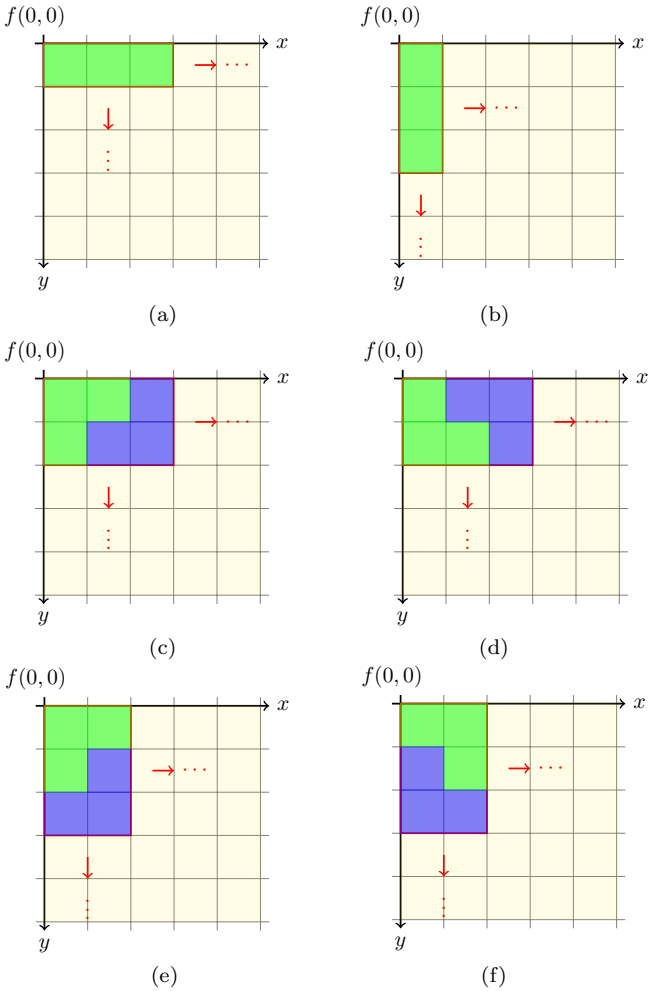


Figure 1: Structures of an image-block of 3-pixels for PG-based RDH scheme: (a) 3×1, (b) 1×3, (c, d) 2×3, and (e, f) 3×2.

block. This processing is abstracted with the function, $block\,(\cdot)$. That is, $B_n = \{b_n^i, b_n^{i+1}, b_n^{i+2}\}$ with $i \in \{1, 2, \cdots, M \times N\}$ for $n \in \{1, 2, \cdots, \frac{M \times N}{3}\}$.

Step 2: A set of sorted image-blocks, $P = \{P_n\}$ is obtained by sorting the pixel-values of each image-block, B_n. For example, a sorted image-block, P_n is obtained by applying the sorting function $sort\,(\cdot)$ block-wise for each B_n. That is, $P_n = \{p_n^i, p_n^{i+1}, p_n^{i+2}\}$, where $p_n^i \leq p_n^{i+1} \leq p_n^{i+2}$.

Step 3: A set of predicted errors E_n is obtained for each sorted block P_n using the function $predict\,(\cdot)$. That is, for each P_n, predicted error

Algorithm 1 PVO Embedding

Input: image I, block-size σ, and payload D
Output: embedded image I'

$\{B_n\} \leftarrow block\,(I, \sigma)$ ▷ n is total no. of blocks
for all B_n **do**
  $P_n \leftarrow sort\,(B_n)$
  $E_n \leftarrow predict\,(P_n)$
  $P'_n \leftarrow embed\,(P_n, E_n, \{d\})$
  $B'_n \leftarrow inverse\_sort\,(P'_n)$
end for
$I' \leftarrow de\_block\,(B'_n)$

$E_n = \{e_n^{max}, e_n^{min}\}$ of the n-th block is obtained using (1).

$$e_n^{max} = p_n^{i+2} - p_n^{i+1} \tag{1a}$$

$$e_n^{min} = p_n^{i} - p_n^{i+1} \tag{1b}$$

Step 4: A pair of predicted errors, e_n^{max} and e_n^{min} of an n-th block is expanded according to the secret bits, $\{d\} \in D$ or is shifted by unit value using (2) and (3) to obtain the modified errors, $\hat{e}_n^{max}$ and $\hat{e}_n^{min}$. These modified errors are then used to compute the set of estimated pixels, $P'_n = \{p_n^{'i}, p_n^{i+1}, p_n^{'i+2}\}$ using (4).

$$\hat{e}_n^{max} = \begin{cases} e_n^{max}, & \text{for } e_n^{max} = 0 \\ e_n^{max} + d, & \text{for } e_n^{max} = 1 \\ e_n^{max} + 1, & \text{for } e_n^{max} > 1 \end{cases} \tag{2}$$

$$\hat{e}_n^{min} = \begin{cases} e_n^{min}, & \text{for } e_n^{min} = \ \ 0 \\ e_n^{min} - d, & \text{for } e_n^{min} = -1 \\ e_n^{min} - 1, & \text{for } e_n^{min} < -1 \end{cases} \tag{3}$$

$$p_n^{'i+2} = p_n^{i+1} + \hat{e}_n^{max} \tag{4a}$$

$$p_n^{'i} = p_n^{i+1} + \hat{e}_n^{min} \tag{4b}$$

Step 5: The embedded pixels of each block are then relocated to their original locations using the inverse of $sort\,(\cdot)$ that we call here $inverse\_sort\,(\cdot)$.

Step 6: The embedded image-blocks are finally combined to return the complete embedded image, I'.

Algorithm 2 PVO Extraction

Input: embedded image I'
Output: original image I and extracted payload D

Initialize: $D \leftarrow \varnothing$
$\sigma \leftarrow blocksize\,(I')$
$\{B'_n\} \leftarrow block\,(I', \sigma)$
for all B'_n **do**
  $P'_n \leftarrow sort\,(B'_n)$
  $E'_n \leftarrow predict\,(P'_n)$
  $(P_n, \{d\}) \leftarrow extract\,(P'_n, E'_n)$
  $D \leftarrow concat\,(D, \{d\})$
  $B_n \leftarrow inverse\_sort\,(P_n)$
end for
$I \leftarrow de\_block\,(B_n)$

3.3 PG-based extraction

PG-based extraction follows the inverse processing of embedding (see Algorithm 2). This algorithm takes the embedded image, I' and block-size, σ as inputs to return the original image, I and extracted data, D. Key steps of this algorithm are briefly discussed below.

Step 1: The extracted payload, D is initialized with an empty array, $\varnothing$.

Step 2: The size of the embedded image-blocks, σ is extracted from I' using $blocksize\,(\cdot)$.

Step 3: A set of image-blocks, $B' = \{B'_n\}$ is obtained from the embedded image, I' using the same function, $block\,(\cdot)$, and σ used in embedding, where B'_n is the n-th image-block of three pixels.

Step 4: A set of sorted image-blocks, P' is obtained from B'. This means that the n-th embedded image-block, P'_n is obtained by the block-wise sorting function $sort\,(\cdot)$ for each B'_n such that $P'_n = \{p_n^{'i}, p_n^{'i+1}, p_n^{'i+2}\}$, where $p_n^{'i} \leq p_n^{'i+1} \leq p_n^{'i+2}$.

Step 5: For each sorted image-block, $P'_n \in P'$, the function $predict\,(\cdot)$ outputs a set of predicted errors, $E'_n = \{\hat{e}_n^{max}, \hat{e}_n^{min}\}$ using (5).

$$\hat{e}_n^{max} = p_n^{'i+2} - p_n^{'i+1} \tag{5a}$$

$$\hat{e}_n^{min} = p_n^{'i} - p_n^{'i+1} \tag{5b}$$

Step 6: From each embedded block, P'_n, the embedded bits, $\{d\}$ are extracted, and the pair of embedded/expanded predicted errors, $\hat{e}_n^{max}$ and $\hat{e}_n^{min}$ are

computed using (6). These errors are then used to compute the originally sorted image-block, $P_n = \{p_n^i, p_n^{i+1}, p_n^{i+2}\}$ using (7). We note that this extraction function is computationally inverse of the embedding function such that $extract\,(\cdot) = embed^{-1}\,(\cdot)$.

$$d = \begin{cases} \hat{e}_n^{max} - 1, & \text{for } 1 \leq \hat{e}_n^{max} \leq 2 \\ -\hat{e}_n^{min} - 1, & \text{for } -2 \leq \hat{e}_n^{min} \leq -1 \end{cases} \tag{6}$$

$$p_n^{i+2} = \begin{cases} p_n^{'i+2}, & \text{for } \hat{e}_n^{max} = 0 \\ p_n^{'i+2} - d, & \text{for } 1 \leq \hat{e}_n^{max} \leq 2 \\ p_n^{'i+2} - 1, & \text{for } \hat{e}_n^{max} > 2 \end{cases} \tag{7a}$$

$$p_n^{i+1} = p_n^{'i+1} \tag{7b}$$

$$p_n^{i} = \begin{cases} p_n^{'i}, & \text{for } \hat{e}_n^{min} = 0 \\ p_n^{'i} + d, & \text{for } -2 \leq \hat{e}_n^{min} \leq -1 \\ p_n^{'i} + 1, & \text{for } \hat{e}_n^{min} < -2 \end{cases} \tag{7c}$$

Step 7: The extracted bits, $\{d\}$ from each embedded image-block is then concatenated with D, which was initialized as an empty array in Step 1.

Step 8: The pixels in each sorted image-block, P_n are relocated to their original locations to obtain the image-block, B_n.

Step 9: Each image-block, B_n is then combined using the function, $de\_block\,(\cdot)$ to obtain the original image, I.

4 Experimental results

The performance of the proposed PG-based RDH scheme has been evaluated and compared with Jung's PG-based scheme [10]. The USC-SIPI test-images [26] of size $256 \times 256 \times 8$ have been used for this performance evaluation. The embedding-capacity and embedding-rate have been determined in terms of the total embedded bits and bit-per-pixels (*bpp*), respectively. For embedding, a set of pseudo-random bits is generated as D. The proposed scheme is implemented using MATLAB R2016b.

Additionally, the embedded image quality has been determined in terms of two popular objective visual quality metrics, *peak signal to noise ratio* (PSNR) defined in (8) and *structural similarity* (SSIM) [28] defined in (9). Here, $M \times N$ is the image size, and $I(i,j)$ and $I'(i,j)$ are the pixel-values of the location (i,j) in an original image and its embedded version, respectively. In (9), μ_x and μ_x' are the average-values of x and x', where $x \in I$ and $x' \in I'$ are the pixels of original and embedded images, respectively. Similarly, σ_x^2 and $\sigma_{x'}^2$ are the variances of x and

x', respectively; $\sigma_{xx'}$ is the covariance of x and x'; c_1 and c_2 are two regularization constants, and L is the dynamic range of the pixel values.

$$\text{MSE} = \frac{\sum_{j=1}^{N}\sum_{i=1}^{M}\left(I'(i,j) - I(i,j)\right)^2}{MN} \tag{8a}$$

$$\text{PSNR} = 10\log\frac{L^2}{\text{MSE}} \tag{8b}$$

$$\text{SSIM} = \frac{(2\mu_x\mu_{x'} + c_1)(2\sigma_{x,x'} + c_2)}{(\mu_x^2 + \mu_{x'}^2 + c_1)(\sigma_x^2 + \sigma_{x'}^2 + c_2)} \tag{9}$$

A better embedding rate-distortion performance has been observed for PVO embedding with L-shaped image-blocks. The pixel-correlations in an image-block thus can be better utilized in PG-based embedding with blocks of size 2×3 or 3×2, resulting in better embedding rate-distortion performance, as illustrated in Table 1. In other words, the room for embedding more bits with the complex image-blocks is mainly resulting from the increasing possibility of expanding the required predicted errors for data-bit embedding as defined with the middle-cases of (2) and (3) in Sec. 3.2, which is attained in the cases of L-shaped image-blocks. For example, the total embedding capacity of Jung's Scheme is 44992 bits (or 0.1716 bpp) for Airplane image, which is increased to 46547 *bits*, 46612 *bits*, and 46762 *bits* (or 0.1776 *bpp*, 0.1778 *bpp*, and 0.1784 *bpp*) for the image-blocks of sizes 3×1, 2×3, and 3×2 of the proposed schemes, respectively.

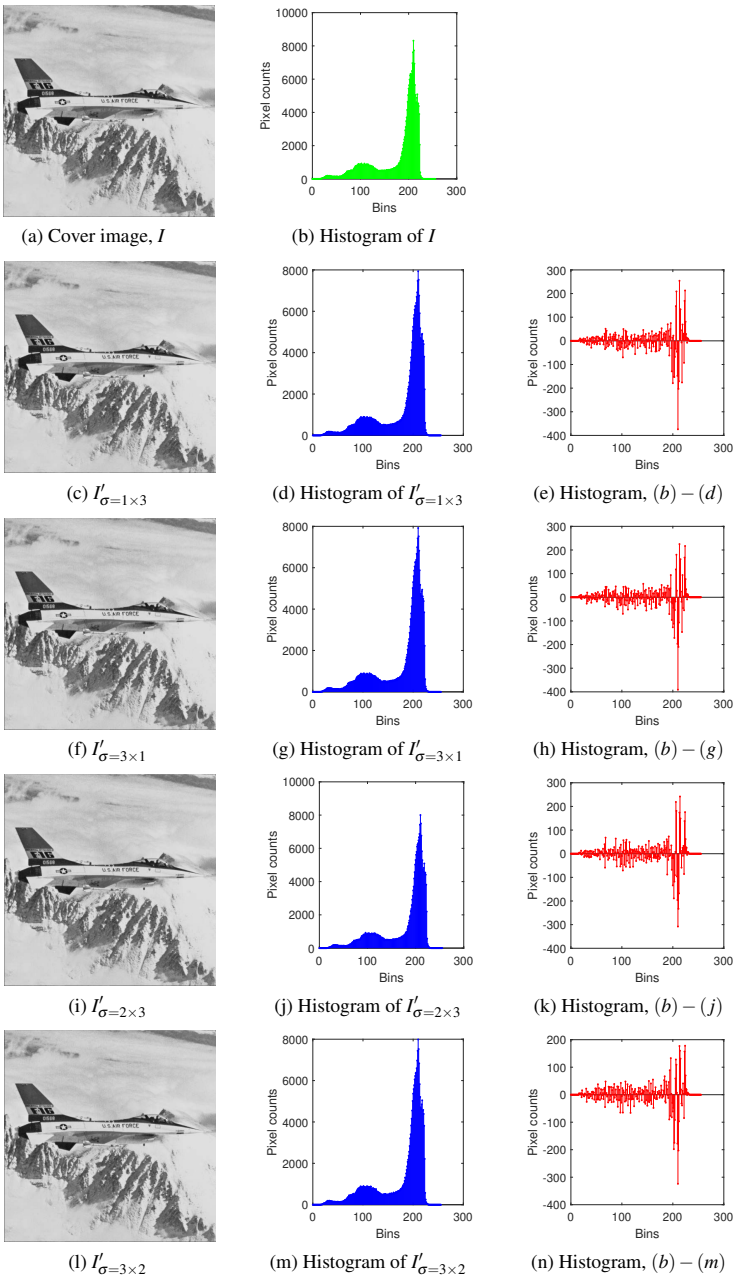
Additionally, the visual quality of the embedded images has remained at a similar level, as evident in Table 1 and Table 2.improved embedding capacity also For example, the PSNR and SSIM values of Airplane embedded images are 51.576 *dB* and 0.9759, respectively. In contrast, the proposed embedding with 3×1, 2×3, and 3×2 offered the PSNR and SSIM values of 51.617 *dB* and 0.9756, 51.639 *dB* and 0.9760, and 51.629 *dB* and 0.9759, respectively. We have observed that, while the performance of the proposed scheme with 3×1 block-size slightly improves over the Jung's scheme, this improvement becomes more noticeable for the other proposed block-sizes (*i.e.*, 2×3 and 3×2). This is because these image-blocks capture pixels in the horizontal, vertical, and diagonal directions to be more correlated than the image-block of size 3×1 (proposed) and 1×3 (Jung's)).

Despite the improvement in the embedding rate, the proposed scheme retains similar intensity distribution of the cover image. The histograms of the cover image and its embedded versions with different values of σ are illustrated in Fig. 2–3. The difference between the cover and any embedded image can hardly be perceived; however, the differences of respective histograms illustrate the changes made in the intensity distribution of the cover image (see the *third-column* from the *left* in Fig. 2–3). Such trivial visual changes remain unnoticeable, as also suggested by the absolute-difference images on the *right-most* columns in those figures.

The above trend of improvement also holds for the average performance of the proposed scheme. The average embedding capacity achieved with the 3×2 size

Table 1: Comparison of rate-distortion performance

Images	**Metric**	**Jung [10]**	**Ours**		
		$(\mathbf{1 \times 3})$	$(\mathbf{3 \times 1})$	$(\mathbf{2 \times 3})$	$(\mathbf{3 \times 2})$
Airfield	Capacity (bits)	27307	29414	30333	30104
	bpp	0.1042	0.1122	0.1157	0.1148
	PSNR (dB)	50.756	50.842	50.864	50.862
	SSIM	0.9941	0.9943	0.9943	0.9943
Airplane	Capacity (bits)	44992	46547	46612	46762
	bpp	0.1716	0.1776	0.1778	0.1784
	PSNR (dB)	51.576	51.617	51.639	51.629
	SSIM	0.9759	0.9756	0.9760	0.9759
Baboon	Capacity (bits)	13226	14046	14090	14087
	bpp	0.0505	0.0536	0.0537	0.0537
	PSNR (dB)	50.263	50.283	50.282	50.286
	SSIM	0.9977	0.9977	0.9977	0.9977
Boat	Capacity (bits)	26588	25521	26224	26338
	bpp	0.1014	0.0973	0.1000	0.1005
	PSNR (dB)	50.681	50.6485	50.660	50.666
	SSIM	0.9926	0.9926	0.9925	0.9925
Couple	Capacity (bits)	34494	34968	34882	34596
	bpp	0.1316	0.1334	0.1331	0.1320
	PSNR (dB)	51.016	51.008	50.996	50.985
	SSIM	0.9916	0.9915	0.9915	0.9915
Elaine	Capacity (bits)	23306	23997	24392	24304
	bpp	0.0889	0.0915	0.0930	0.0927
	PSNR (dB)	50.595	50.612	50.633	50.629
	SSIM	0.9929	0.9926	0.9928	0.9928
Goldhill	Capacity (bits)	27021	29365	28280	28573
	bpp	0.1031	0.1120	0.1079	0.1090
	PSNR (dB)	50.688	50.759	50.719	50.730
	SSIM	0.9922	0.9924	0.9923	0.9923
Peppers	Capacity (bits)	33483	31933	33423	33802
	bpp	0.1277	0.1218	0.1275	0.1289
	PSNR (dB)	50.923	50.869	50.914	50.916
	SSIM	0.9887	0.9885	0.9886	0.9886
Tiffany	Capacity (bits)	41750	38807	41864	41680
	bpp	0.1593	0.1480	0.1597	0.1590
	PSNR (dB)	51.316	51.183	51.305	51.303
	SSIM	0.9829	0.9826	0.9829	0.9829



(a) Cover image, I (b) Histogram of I

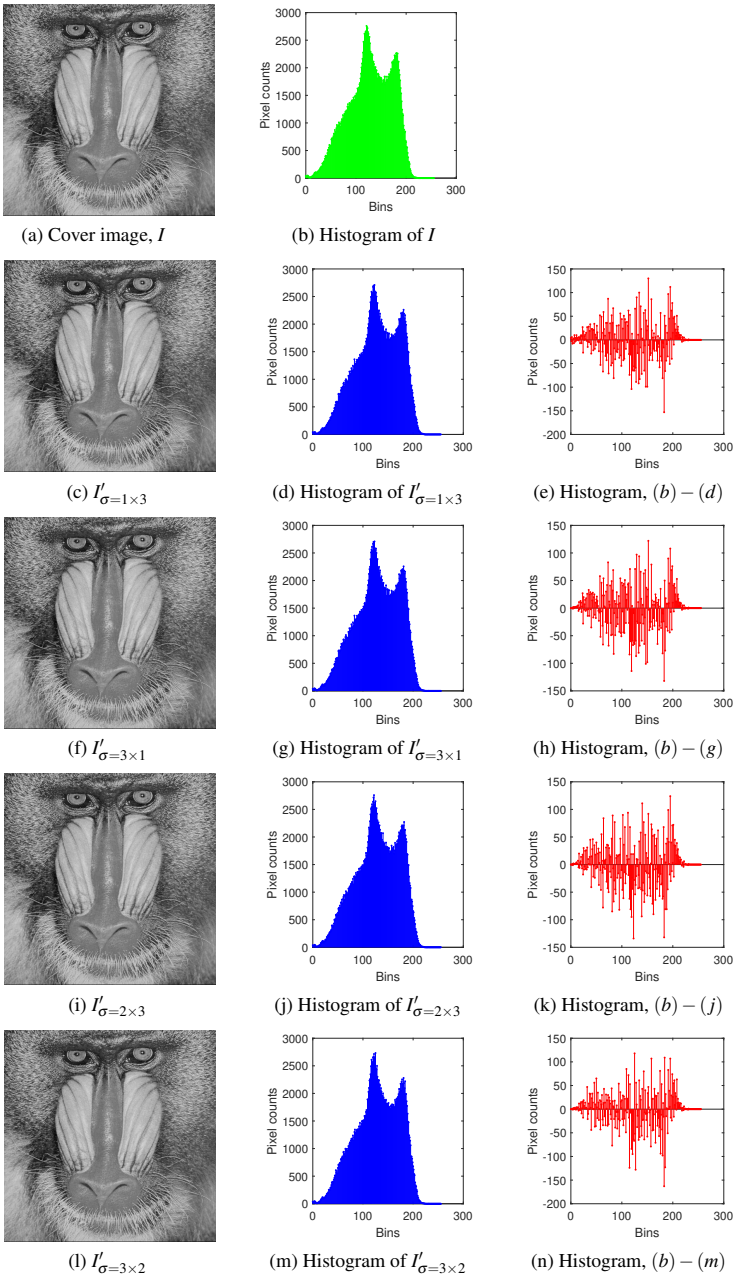
(c) $I'_{\sigma=1\times3}$ (d) Histogram of $I'_{\sigma=1\times3}$ (e) Histogram, $(b)-(d)$

(f) $I'_{\sigma=3\times1}$ (g) Histogram of $I'_{\sigma=3\times1}$ (h) Histogram, $(b)-(g)$

(i) $I'_{\sigma=2\times3}$ (j) Histogram of $I'_{\sigma=2\times3}$ (k) Histogram, $(b)-(j)$

(l) $I'_{\sigma=3\times2}$ (m) Histogram of $I'_{\sigma=3\times2}$ (n) Histogram, $(b)-(m)$

Figure 2: Comparison of the cover image and its histogram with different embedded versions and their histograms for the *Airplane* image.



(a) Cover image, I (b) Histogram of I

(c) $I'_{\sigma=1\times3}$ (d) Histogram of $I'_{\sigma=1\times3}$ (e) Histogram, $(b)-(d)$

(f) $I'_{\sigma=3\times1}$ (g) Histogram of $I'_{\sigma=3\times1}$ (h) Histogram, $(b)-(g)$

(i) $I'_{\sigma=2\times3}$ (j) Histogram of $I'_{\sigma=2\times3}$ (k) Histogram, $(b)-(j)$

(l) $I'_{\sigma=3\times2}$ (m) Histogram of $I'_{\sigma=3\times2}$ (n) Histogram, $(b)-(m)$

Figure 3: Comparison of the cover image and its histogram with different embedded versions and their histograms for the *Baboon* image.

Table 2: Comparison of average rate-distortion performance

Metric	**Jung [10]**	**Ours**		
	$(\mathbf{1 \times 3})$	$(\mathbf{3 \times 1})$	$(\mathbf{2 \times 3})$	$(\mathbf{3 \times 2})$
Capacity (bits)	31223	31387	32228	32191
bpp	0.1191	0.1197	0.1229	0.1228
PSNR (dB)	50.921	50.916	50.948	50.944
SSIM	0.9891	0.9890	0.9891	0.9891

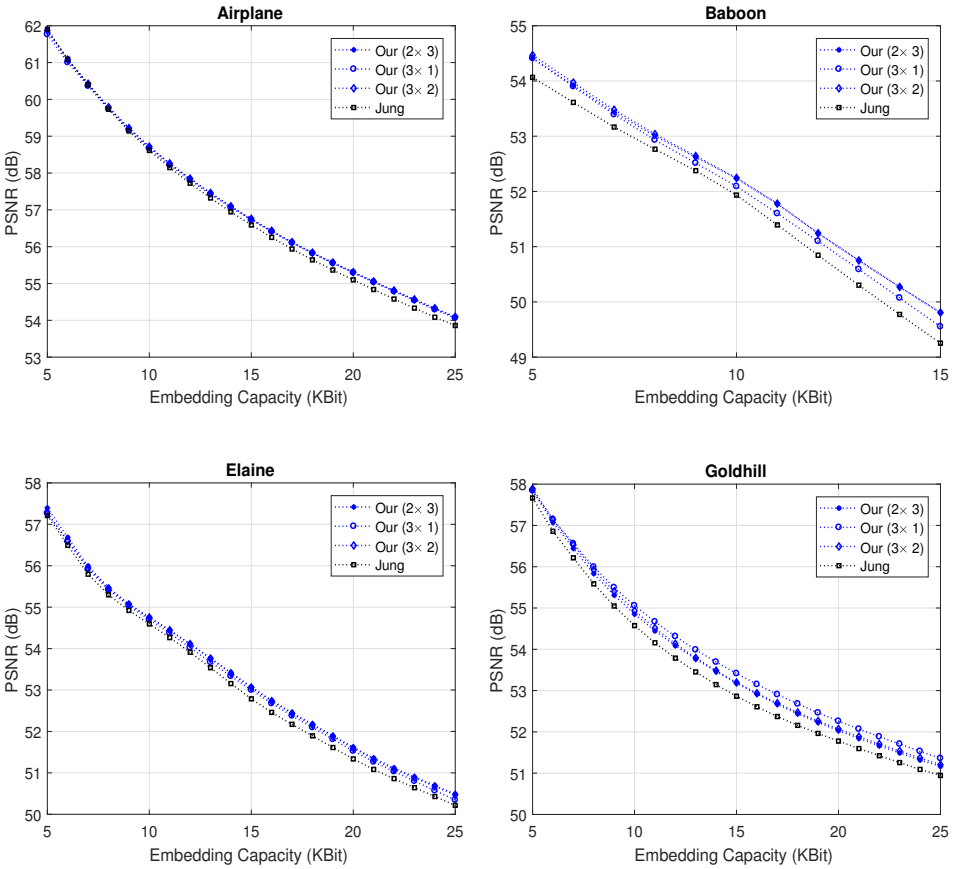


Figure 4: Embedding rate-distortion performance comparison

image-block is 32191 *bits*, and that with an image-block of size 2×3 is 32228 *bits*; whereas, the capacity is found of 31223 *bits* and 31387 *bits* for the image-blocks of size 1×3 and 3×1, respectively. This improved embedding capacity also maintains an improved average PSNR and similar SSIM values in case of the image-block of size 2×3. We note that similar improvements in the rate-distortion performance of the proposed RDH scheme also exist for the other test images we experimented with.

5 Conclusions

PG-based RDH is generalized for different image-blocks and its embedding rate-distortion performance is investigated for better utilization of block-pixels correlation. The image-blocks with different structures have been investigated for the PG-based embedding. The presented simulation and experimental results in this paper suggest that a better rate-distortion performance can be obtained with the embedding in an L-shaped image-block capturing pixels in the horizontal, vertical, and diagonal contexts. In other words, the PG-based embedding with 2×3 and 3×2 image-blocks would offer an improved rate-distortion performance compared to the other block-sizes and the Jung's scheme. This consideration of constructing image-block may also contribute to the development of PG-based RDH schemes in the future.

References

[1] Alattar, A. M. Reversible watermark using the difference expansion of a generalized integer transform. *IEEE Transactions on Image Processing*, 13(8):1147–1156, 2004. DOI: 10.1109/TIP.2004.828418.

[2] Chen, X., Li, X., Yang, B., and Tang, Y. Reversible image watermarking based on a generalized integer transform. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2382–2385, 2010. DOI: 10.1109/ICASSP.2010.5496175.

[3] Coatrieux, G., Pan, W., Cuppens-Boulahia, N., Cuppens, F., and Roux, C. Reversible watermarking based on invariant image classification and dynamic histogram shifting. *IEEE Transactions on Information Forensics and Security*, 8(1):111–120, 2013. DOI: 10.1109/TIFS.2012.2224108.

[4] Coltuc, D. Improved embedding for prediction-based reversible watermarking. *IEEE Transactions on Information Forensics and Security*, 6(3):873–882, 2011. DOI: 10.1109/TIFS.2011.2145372.

[5] Coltuc, D. and Chassery, J. Very fast watermarking by reversible contrast mapping. *IEEE Signal Processing Letters*, 14(4):255–258, 2007. DOI: 10.1109/LSP.2006.884895.

[6] Eskicioglu, A. M. Multimedia security in group communications: recent progress in key management, authentication, and watermarking. *Multimedia Systems*, 9(3):239–248, 2003. DOI: 10.1007/s00530-003-0095-2.

[7] Hasib, S. A. and Nyeem, H. Developing a pixel value ordering based reversible data hiding scheme. In *2017 3rd International Conference on Electrical Information and Communication Technology (EICT)*, pages 1–6, 2017. DOI: 10.1109/EICT.2017.8275160.

[8] Hasib, S. A. and Nyeem, H. Rate-distortion analysis of an improved pixel value ordering based reversible embedding. In *2017 2nd International Conference on Electrical Electronic Engineering (ICEEE)*, pages 1–4, 2017. DOI: 10.1109/CEEE.2017.8412855.

[9] Jun Tian. Reversible data embedding using a difference expansion. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(8):890–896, 2003. DOI: 10.1109/TCSVT.2003.815962.

[10] Jung, Ki-Hyun. A high-capacity reversible data hiding scheme based on sorting and prediction in digital images. *Multimedia Tools and Applications*, 76(11):13127–13137, 2017. DOI: 10.1007/s11042-016-3739-x.

[11] Li, X., Yang, B., and Zeng, T. Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. *IEEE Transactions on Image Processing*, 20(12):3524–3533, 2011. DOI: 10.1109/TIP.2011.2150233.

[12] Li, Xiaolong, Li, Jian, Li, Bin, and Yang, Bin. High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. *Signal Processing*, 93(1):198 – 205, 2013. DOI: https://doi.org/10.1016/j.sigpro.2012.07.025.

[13] Ni, Zhicheng, Shi, Yun-Qing, Ansari, N., and Su, Wei. Reversible data hiding. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(3):354–362, 2006. DOI: 10.1109/TCSVT.2006.869964.

[14] Nyeem, H., Boles, W., and Boyd, C. Developing a digital image watermarking model. In *2011 Int. Conf. on Digital Image Computing: Techniques and Applications*, pages 468–473, 2011. DOI: 10.1109/DICTA.2011.85.

[15] Nyeem, H.A. *A digital watermarking framework with application to medical image security.* PhD thesis, Queensland University of Technology, 2014.

[16] Nyeem, Hussain, Boles, Wageeh, and Boyd, Colin. Digital image watermarking: its formal model, fundamental properties and possible attacks. *EURASIP Journal on Advances in Signal Processing*, 2014(1):135, 2014. DOI: 10.1186/1687-6180-2014-135.

[17] Nyeem, Hussain, Boles, Wageeh, and Boyd, Colin. Content-independent embedding scheme for multi-modal medical image watermarking. *Biomedical engineering online*, 14(1):1–19, 2015. DOI: 10.1186/1475-925X-14-7.

[18] Nyeem, Hussain, Boles, Wageeh, and Boyd, Colin. Watermarking capacity control for dynamic payload embedding. In *Recent Advances in Information and Communication Technology 2015*, pages 143–152. Springer, 2015. DOI: 10.1007/978-3-319-19024-2-15.

[19] Ou, Bo, Li, Xiaolong, and Wang, Jinwei. High-fidelity reversible data hiding based on pixel-value-ordering and pairwise prediction-error expansion. *Journal of Visual Communication and Image Representation*, 39:12–23, 2016. DOI: 10.1016/j.jvcir.2016.05.005.

[20] Ou, Bo, Li, Xiaolong, Zhao, Yao, and Ni, Rongrong. Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion. *Signal processing: image communication*, 29(7):760–772, 2014. DOI: 10.1016/j.image.2014.05.003.

[21] Peng, F., Li, X., and Yang, B. Improved PVO-based reversible data hiding. *Digital Signal Processing*, 25:255–265, 2014. DOI: 10.1016/j.dsp.2013.11.002.

[22] Qu, Xiaochao and Kim, Hyoung Joong. Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding. *Signal Processing*, 111:249–260, 2015. DOI: 10.1016/j.sigpro.2015.01.002.

[23] Sachnev, V., Kim, H. J., Nam, J., Suresh, S., and Shi, Y. Q. Reversible watermarking algorithm using sorting and prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(7):989–999, 2009. DOI: 10.1109/TCSVT.2009.2020257.

[24] Shi, Y., Li, X., Zhang, X., Wu, H., and Ma, B. Reversible data hiding: Advances in the past two decades. *IEEE Access*, 4:3210–3237, 2016. DOI: 10.1109/ACCESS.2016.2573308.

[25] Thodi, D. M. and Rodriguez, J. J. Expansion embedding techniques for reversible watermarking. *IEEE Transactions on Image Processing*, 16(3):721–730, 2007. DOI: 10.1109/TIP.2006.891046.

[26] USC-SIPI. Image database. http://sipi.usc.edu/database/.

[27] Wahed, M. A. and Nyeem, H. Developing a block-wise interpolation based adaptive data embedding scheme. In *2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, pages 1–6, 2016. DOI: 10.1109/CEEICT.2016.7873152.

[28] Zhou Wang, Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. DOI: 10.1109/TIP.2003.819861.

Received 2nd October 2018