Journal of Information Hiding and Multimedia Signal Processing ©2024 ISSN 2073-4212
Ubiquitous International Volume **15**, Number **4**, December **2024**

An Improved Difference Expansion-Based Reversible Image Watermarking Algorithm with No Location Map

Tarek ABED
ETA Laboratory, Department of Electronics
Mohamed El Bachir El Ibrahimi University of Bordj Bou Arreridj
Bordj Bou Arreridj, Algeria
tarek.abed@univ-bba.dz

Saad BOUGUEZEL
CCNS Laboratory, Department of Electronics
University Ferhat Abbas Setif-1
Setif, Algeria
bouguezel_saad@yahoo.com

Khaled ROUABAH
Electronics Department
University of M'sila, University Pole
Road of Bordj Bou Arréridj, M'sila 28000, Algeria
Khaled.rouabah@univ-msila.dz

Received July 15, 2024, revised September 26, 2024, accepted September 28, 2024.

Abstract. *In this paper, we propose a new reversible image watermarking algorithm based on Difference Expansion (DE) approach. In the developed algorithm, we completely eliminate the location map by introducing an appropriate selection of pixel pairs from the smoothest regions of the image using a special case expandability criterion. It should be mentioned that, in the case of an image of any size, this elimination of the location map, which is considered to be a major drawback of the existing DE-based data hiding techniques, costs the proposed algorithm just few additional bits for the decoding. For instance, for a* 512×512 *8-bit gray-scale image, the proposed algorithm only requires* 57 *bits to be embedded in the host image for perfect recovering. This can dramatically increase the size of the pure payload. Furthermore, we experimentally show that the proposed algorithm outperforms the existing DE-based data hiding methods in terms of Peak Signal to Noise Ratio (PSNR).*

Keywords: Watermarking, Difference Expansion, Reversible Data Hiding, Data Embedding, Location Map.

---

1. **Introduction.** Sharing or transferring images and other digital contents like audio and video via communication networks is now a common practice. Specifically, the images may be used to carry some additional data, which can serve for authentication, as an ownership code or just an annotation. The inclusion of supplement data in an image is called "data hiding" or "data embedding". Robustness, invisibility, and payload are three crucial and opposing properties for any image watermarking systems [1]. In this context, many data hiding techniques have been proposed in the literature to achieve such an embedding and are also called watermarking techniques when data is used for ownership

protection [2,3] and steganography techniques in the case where the image is used to carry secret messages [4, 5]. Data hiding techniques provide an invisible communication of the supplement data, which is called payload. The embedded image, which is called cover image, is then obtained by invisibly embedding this payload into the original image, which is also called stego-image. The existing embedding techniques always cause distortion in the stego-image, even if it is visually imperceptible. Reducing this distortion is then an important concern in designing data hiding methods [6].

The distortion occurred to the host image may be permanent, even after extracting the hidden data. In some special fields, such as valuable art, medical or military applications, this permanent distortion can be a challenging problem and even absolutely intolerable. Thus, the embedding process must be completely reversible, i.e., both original image and hidden data must be restored without any loss during the data extraction (decoding process). For this purpose, many lossless or Reversible Data Hiding (RDH) techniques have been reported in the literature. The earlier methods were proposed by Barton [7] in 1997 and Honsinger et al. [8] in 2001. During the two last decades, many other RDH techniques have been reported in the literature [9–12]. They can be classified into five main categories, namely lossless compression [13,14], pixel histogram modification [15–17], Pixel Value Ordering (PVO) [18–22], Interpolation Technology (IT) [23] and DE [24–33]. However, these RDH techniques require additional data to be embedded with the payload, which increase the distortion and reduce the embedding capacity of the cover image.

One of the most existing simple and efficient embedding approaches is the DE proposed firstly by Tian [24] in 2003. This technique offers high embedding capacity and low distortion in image quality [10], but its major drawbacks are the Location Map (LM) and lack of capacity control [26]. The DE divides the image into non-overlapping pairs of pixels, difference and average of each pair are then computed. The binary representation of the difference is then one-bit left shifted (doubled) to create a free space to embed one bit of information. Not all pairs can be used for data hiding; the only expanded pairs are those that do not cause overflow or underflow. A location map is necessary and used to indicate whether the pairs are expanded or not. The location map is then compressed to free space for payload to be embedded.

Many extensions to Tian's DE have been developed [11]. To increase hiding ability, Alattar [25] proposed a generalized DE using vectors of triplets and quads pixels. Thodi et al. [26] proposed two histogram shifting-based techniques to reduce the amount of embedded auxiliary information. Kim et al. [28] proposed a DE transform with a reduced location map. Thodi et al. [26] also introduced Prediction Error Expansion (PEE). To embed data, PEE uses prediction error for all pixels, so the embedding capacity can be increased up to 1 *bpp* instead of 0.5 *bpp* in Tian's DE. Kamstra et al. [27] used a sorting scheme based on the average image to reduce overhead embedded data. However, all the above-mentioned extensions require a LM and their attempts to reduce its size by compression did not solve the problem.

In this work, we propose a DE-based RDH method that brings a major modification to Tian's DE as it effectively eliminates the location map and necessary compression steps in the decoding process. This is basically accomplished through an appropriate selection of pixel pairs from the smoothest regions of the image using a special case expandability criterion. As a result, a high embedding capacity that is fully controlled is achieved in a single pass embedding at a very low computational effort. For instance, for a 512×512 8-*bit* grayscale image, only 57 *bits* are needed to recover original image and embedded payload. It should be mentioned that this proposed novel approach is completely different than that used in [29], which, to the best of authors knowledge, is the only existing reference that considers the elimination of the LM.

The rest of the paper is organized as follows. Section II reviews the basics of the DE. The proposed method and its corresponding coding and decoding algorithms are developed in Section III. In Section IV, experimental results as well as the comparison with prior RDH methods are presented. The performance of different RDH methods is evaluated by metrics including the stego-image quality (the imperceptibility of distortion), the number of embedding bits (embedding capacity, which is a critical concern for RDH), and the side information (additional data) needed for decoding. Finally, Section V concludes this paper.

2. **DIFFERENCE EXPANSION BASICS.** In the DE, the host image pixels are divided into non-overlapping pairs. Pairs can be chosen horizontally, vertically, or any key-based specific pattern [24]. For each pixel-pair having gray scales x and y, the difference h and average l are computed using integer Haar wavelet transform as:

$$h = x - y, \quad l = \left\lfloor \frac{x+y}{2} \right\rfloor \tag{1}$$

where $0 \leq x, y \leq 255$ for an $8-bit$ grayscale image and the symbol $\lfloor . \rfloor$ denotes the floor function. This transformation is invertible, i.e., x and y can be retrieved from h and l as:

$$x = l + \left\lfloor \frac{h+1}{2} \right\rfloor, \quad y = l - \left\lfloor \frac{h}{2} \right\rfloor \tag{2}$$

The difference h is used for embedding an information bit b. To make space for embedding, the binary value of h is one-bit shifted left, which means that h is doubled, then the bit b is added to h to form a new difference h' as:

$$h' = 2h + b \tag{3}$$

x' and y' are the new grayscale values of x and y in the embedded image and are computed by replacing h by h' in (2). To prevent overflow and underflow, these new values must be restricted in the range of $[0, 255]$ as follows:

$$0 \leq l + \left\lfloor \frac{h'+1}{2} \right\rfloor \leq 255, 0 \leq l - \left\lfloor \frac{h'}{2} \right\rfloor \leq 255 \tag{4}$$

Since h' and l are integers, (4) is equivalent to:

$$|h'| \leq 2(255 - l), |h'| \leq 2l + 1 \tag{5}$$

where $|.|$ denotes the absolute value.

A difference h is expandable if, after embedding a bit b, the expanded difference h' satisfies the condition (5). From (3) and (5), the expandability condition can be formulated as :

$$|2h + b| \leq 2(255 - l), \quad |2h + b| \leq 2l + 1 \tag{6}$$

for both cases $b = 0$ and $b = 1$.

A difference h is changeable if its LSB is modified without causing an overflow or underflow. Thus, the changeability condition can be given in the form:

$$\left| 2 \left\lfloor \frac{h}{2} \right\rfloor + b \right| \leq 2(255 - l), \left| 2 \left\lfloor \frac{h}{2} \right\rfloor + b \right| \leq 2l + 1 \tag{7}$$

for both cases $b = 0$ and $b = 1$.

By using (6) and (7), the differences are classified in three sets: expandable, changeable, and nonchangeable pairs. All expandable pairs are also changeable. After expansion, the changeable pairs will remain changeable, but the expandable pairs can become changeable or remain expandable. The embedding capacity of the DE is, at best (all pairs are expandable), 0.5 *bit* per pixel (*bpp*).

The difference histogram for most natural images is close to a Laplacian distribution; difference values with small magnitudes occur more frequently. Thus, selected locations for expansion embedding from the set of expandable pairs should have small magnitude. These are preferred because the resulting distortion will be smaller than using pairs with high magnitude differences. Not all expandable differences are used for data embedding. Depending on the payload size and the tolerable distortion, a threshold T must be set to select differences for embedding. A location map is then formed to qualify an expandable pair as a selected or a not-selected location for data embedding. The location map without compression needs 0.5 *bpp* to be embedded, which makes it a limiting parameter in the embedding process. It is necessary then to compress it in order to free space for the payload. The compressed location map is concatenated with the payload, and then embedded into the image. The location map is the major Tian's DE drawback; it significantly reduces the embedding capacity and increases distortion. Another drawback of Tian's DE is the lack of capacity control, i.e., it is not possible to determine in advance the embedding capacity because it depends on the compression ratio of the location map. To overcome this limitation, we will give in what follows the principle of our proposed algorithm.

3. **PROPOSED ALGORITHM.**

Tian's DE method has been a very important step in the field of reversible data hiding. It allows embedding of a large amount of data while keeping distortion low. Several methods have been proposed in the literature to overcome the location map and lack of capacity control drawbacks of Tian's DE method. Specifically, most existing Tian's DE extensions seek to reduce the location map [24–33]. However, the embedding capacity is still limited by the location map and its compression performance. In this section, we propose a new algorithm based on Tian's DE. This algorithm has many advantages such as simple structure without complex computations, no need to a location map, no need to a compression step, and a big embedding capacity can be reached in a single pass embedding.

The DE calculates the differences and averages of all pixel-pairs in the image. The differences are modified by data embedding while the averages remain unchanged in the embedded image. In our algorithm, only the averages and two thresholds are used to identify selected expandable locations for the embedding process. For this purpose, the smoothness of the regions of the image is determined using (8), which is introduced in this paper as the criterion to identify the expandable locations. We use the sum of vertical and horizontal gradients of the averages to calculate a smoothness ratio for the region where the pixel-pair belongs. For a pixel-pair with average $L_{i,j}$, the smoothness ratio (SR) given by (8) is computed using neighboring averages in a 3×3 window as shown in Fig. 1.

$$\begin{aligned} SR = {} & |L_{i-1,j-1} - L_{i,j-1}| + |L_{i,j-1} - L_{i+1,j-1}| + |L_{i-1,j} - L_{i,j}| + |L_{i,j} - L_{i+1,j}| + \\ & |L_{i-1,j+1} - L_{i,j+1}| + |L_{i,j+1} - L_{i+1,j+1}| + |L_{i,j+1} - L_{i,j}| + |L_{i-1,j-1} - L_{i-1,j}| + \\ & |L_{i-1,j} - L_{i-1,j+1}| + |L_{i,j-1} - L_{i,j}| + |L_{i,j} - L_{i,j+1}| + |L_{i+1,j-1} - L_{i+1,j}| + \\ & |L_{i+1,j} - L_{i+1,j+1}| \end{aligned} \tag{8}$$

$L_{i-1,j-1}$	$L_{i,j-1}$	$L_{i+1,j-1}$
$L_{i-1,j}$	$L_{i,j}$	$L_{i+1,j}$
$L_{i-1,j+1}$	$L_{i,j+1}$	$L_{i+1,j+1}$

FIGURE 1. 3x3 window pixel-pair neighboring averages

Pixel-pairs belonging to regions with small SR values are likely to be expandable. Alone, this criterion cannot precisely define the locations for embedding. A special case expandability condition (called new expandability in [28]) must be satisfied for the selected locations. This special case condition is derived for a difference threshold T from the expandability condition given by (6) as:

$$T \leq l < 255 - T, |h| \leq T \tag{9}$$

All the locations satisfying the condition given by (9) also satisfy the expandability condition given by (6), but not all expandable pairs satisfy (9). For example, for a pixel-pair with $l = 3$ and $h = 2$, if the threshold is set to $T = 4$, then (6) is satisfied, whereas (9) is not. When using (9) to select pixel-pairs for embedding, some expandable locations will be lost and cannot be used for embedding. Such a situation occurs more when the threshold is set in the high magnitudes of difference values. The use of (9) instead of (6) ensures retrieving these locations at the decoder without needing a location map. Therefore, this is an interesting consequence even at the expense of a reduction in the number of selected expandable locations.

When the value of T is very high, ambiguous changeable pairs may appear. Those are the pairs with h greater than T in the interval $[-2T, 2T + 1]$ and l satisfies (9). These pairs are ambiguous because they will interfere with selected expandable locations in the decoding process. Ambiguous changeable pairs may appear in the case of images with few smooth areas. This situation rarely occurs, so the algorithm must be stopped when the number A of ambiguous changeable pairs becomes greater than zero.

The parameters provided to the decoder are two threshold values and the length of the payload. These thresholds are Smoothness Ratio Threshold (SRT) and difference threshold T, which are embedded in the host image using LSB replacement and then can directly be extracted at the decoding stage. The length P of the payload is provided to the decoder to indicate when data extraction ends.

For an $8-$ *bit* grayscale image, the SRT greatest value is $12 \times 255 = 3060$, so its binary representation needs at most 12 *bits*, and the greatest value of the threshold T is 255 , which needs 8 *bits*. For an N by M image, the total number of pixel-pairs is $(N/2) \times M$, which requires $\lceil log_2 ((N/2) \times M) \rceil$ *bits* for its binary representation, where $\lceil . \rceil$ denotes ceil rounding function. In the case of a 512×512 image, there are 131072 pixel-pairs and the binary representation of the length of the payload is 17 *bits*. Therefore, the total additional data required to be embedded within the original image is only 57 *bits*; 20 *bits* for the thresholds (12 *bits* for SRT and 8 *bits* for T), which are embedded using LSB replacement, and a bitstream α of 37 *bits*, which is constructed by 20 original bits string β of the locations used to embed the thresholds and 17 *bits* string γ of the payload length P . Hence, the bitstream α is formed as:

$$\alpha = \beta \oplus \gamma \tag{10}$$

and concatenated with the payload π to construct the final bitstream ζ of length $37+P$ *bits* as:

$$\zeta = \alpha \oplus \pi \tag{11}$$

where the symbol $\oplus$ denotes string concatenation. The bitstream ζ is then embedded using the DE process proposed in the next subsection for which the embedded differences in the stego-image, originally in the range of $[-T, +T]$, will be extended in the range of $[-2T, 2T+1]$. The corresponding pixel-pairs will preserve the same smoothness ratio and their averages l will be also preserved and still satisfy (9).

3.1. **Proposed encoding algorithm.** For a given payload π with length P , the proposed encoding algorithm consists of the following steps:

1. Group the host image in non-overlapping pixel-pairs in a given pattern such as row-wise, column-wise or others,
2. Compute differences and averages of all pixel-pairs,
3. Compute smoothness ratio for all pixel-pairs using (8),
4. Mark the first 20 changeable locations satisfying the condition in (6) or (7),
5. Select embedding locations using the procedure below:

```
  Set $SRT = -1$
  do
  {   $SRT = SRT + 1$
      Determine all expandable pairs using (6) with a smoothness ratio less or equal to $SRT$
      Set T = maximum difference value for those expandable pairs
      Select among those expandable pairs all pairs satisfying (9), where their number is noted $EP$
      Calculate the number $A$ of ambiguous changeable pairs
  }
  while $EP < (P + 37)$ and $A = 0$
  if $A > 0$, then set $SRT = SRT - 1$ and recalculate its corresponding value of $T$
```

6. Save the 20 LSB's of the locations marked in Step 4 in string β and the payload length P in string γ,
7. Construct the bitstreams α and then ζ using (10) and (11), respectively,
8. Embed *SRT* and T thresholds using LSB modification in the locations marked in Step 4,
9. Embed the bitstream ζ using DE,
10. Compute the stego-image.

3.2. **Proposed decoding algorithm.** The proposed decoder only needs the average values, *SRT* and T thresholds, and the payload length P to detect the locations that have been used for embedding. The differences h obtained using those locations are in the range $[-2T, 2T+1]$ and the smoothness ratios *SR* are less or equal to *SRT*. The corresponding averages l must all satisfy (9) because T was found as the greatest difference value for expandable pairs having smoothness ratios less or equal to *SRT*.

The difference obtained from a pixel-pair that has been employed for embedding using LSB replacement can be restored by:

$$h = 2\left\lfloor \frac{h'}{2} \right\rfloor + b, \tag{12}$$

where b is the original LSB of the difference h.

The difference obtained from of a pixel-pair that has been employed for embedding using DE can be restored by:

$$h = 2\left\lfloor \frac{h'}{2} \right\rfloor \tag{13}$$

The proposed decoding process is achieved by adopting the following steps:

1. Group embedded image pixels in non-overlapping pairs in the same pattern as in the encoder,
2. Compute differences and averages of all pixel-pairs,
3. Compute smoothness ratio for all pixel-pairs using (8),
4. Extract LSB's from the first 20 changeable locations satisfying (6) or (7),
5. Obtain SRT and T values,
6. Start the extraction process after skipping the first 20 changeable pairs,
7. Extract the bitstream α of 37 *bits* using reverse DE from the locations satisfying the following conditions:
   - Average l satisfy (9)
   - Difference h in the range $[-2T, 2T+1]$
   - Smoothness ratio less or equal to SRT
8. Compute the value of the payload length P using the last 17 *bits* of α extracted in Step 7,
9. After skipping the 37 locations used in Step 7, start extracting the payload π from the next P locations satisfying the three conditions of Step 7,
10. Restore the differences of the locations employed in Step 4 using (12) and the 20 LSB's extracted in Step 7,
11. Restore the differences of the $37 + P$ locations employed in Steps 7 and 9 using (13),
12. Restore the host image.

4. **EXPERIMENTAL RESULTS.** In this section, we implement the proposed RDH method in C++ and evaluate its performance on commonly used $8-bit$ grayscale 512×512 Airplane, Lena, and Mandrill images. The embedding and decoding experiments are achieved by adopting horizontal and vertical pairing patterns and carried out using a payload randomly generated by C++ random function. The proposed method is then compared with existing DE-based reversible data hiding methods, specifically those of Thodi et al. [26], Kams et al. [27], Kim et al. [28] and Lin et al. [29]. Moreover, the performance of the proposed method is compared to that of the recently reported work of Gujjunoori et al. [30] in the case of some miscellaneous images of the USCSIPI image database. The performance of different methods is evaluated using the Embedding Capacity *(EC)*, *(PSNR)* and Structural Similarity Index Measure *(SSIM)* metrics.

For a one-pass embedding process, the results for maximum embedding capacity *(Max EC)* and its corresponding embedding rate *(ER)* and *(PSNR)* are given in Table 1.

Table 1. Maximum embedding capacity for different images and pairing patterns

Image / Pairing	Airplane			Lena			Mandrill		
	Max EC	*ER (bpp)*	*PSNR*	*Max EC*	*ER (bpp)*	*PSNR*	*Max EC*	*ER (bpp)*	*PSNR*
Vertical	66106	0.25	46.17	116956	0.45	39.97	72218	0.28	31.39
Horizontal	100081	0.38	44.21	99373	0.38	41.80	104123	0.40	31.74

Tables 2, 3 and 4 show the *PSNR* obtained by the proposed method and those reported in [26], [27], [28] and [29] for different images and various embedding rates. It is seen from these tables that the proposed method outperforms the existing methods in the case of Airplane image and presents excellent results in the cases of Lena and Mandrill images.

Table 2. *PSNR* obtained by different methods for Airplane image and various embedding rates

ER (bpp)	Thodi *et al.* [26]	Kams *et al.* [27]	Kim *et al.* [28]	Lin *et al.* [29]	Proposed algorithm
0.05		59.22	59.35	59.25	58.92
0.1		55.42	55.27	55.23	55.07
0.15	51.52	52.87	52.71	52.64	52.85
0.2	49.85	50.99	50.82	50.84	51.01
0.25	48.32	49.29	49.15	49.22	49.49
0.3	46.65	47.76	47.47	47.54	47.89
0.35	44.85	45.66	45.43	45.68	45.87
0.4	42.76	42.93	42.35	43.21	
0.45	39.7	37.88		39.73	

Table 3. *PSNR* obtained by different methods for Lena image and various embedding rates

ER (bpp)	Thodi *et al.* [26]	Kams *et al.* [27]	Kim *et al.* [28]	Lin *et al.* [29]	Proposed algorithm
0.05		53.26	53.84	54	55.95
0.1	51.64	50.08	50.78	50.51	52.55
0.15	49.95	48.05	48.39	48.41	50.52
0.2	47.96	46.57	46.83	46.6	48.94
0.25	46.23	45.22	45.21	45.28	47.68
0.3	44.69	43.8	43.79	44.02	46.43
0.35	43.27	42.21	41.93	42.87	45.02
0.4	41.66	40.29	39.44	41.37	43.39
0.45	39.96	37.85		39.32	39.98

Table 4. *PSNR* obtained by different methods for Mandrill image and various embedding rates

ER (bpp)	Thodi *et al.* [26]	Kams *et al.* [27]	Kim *et al.* [28]	Lin *et al.* [29]	Proposed algorithm
0.05	49.30	47.37	47.89	49.80	53.66
0.1	44.19	43.18	43.81	44.93	49.14
0.15	41.18	40.29	40.62	42.09	45.62
0.2	38.60	37.51	37.74		42.57
0.25	36.29	35.14	34.92		38.93
0.3	34.52	33.04	32.51		36.61
0.35	32.69	31.46			34.23
0.4	30.97	29.98			31.74
0.45	29.09	28.79			

In addition, it should be noted that in our method the embedding capacity is totally dedicated to the pure payload, while in the other schemes, except the one reported in [29], the embedding capacity contains pure payload and location map. Furthermore, the computational complexity of the proposed algorithm is reduced due to the fact that the existing schemes require a location map compression step, which is not the case with our algorithm.

Let us now compare the performance of the proposed method with that of the schemes recently reported in [30]. The comparison is performed in terms of*PSNR* and *SSIM* metrics and the corresponding results are presented in Tables 5 and 6 for some miscellaneous images of the USC-SIPI image database. In these tables:

DER: Difference expansion-based reversible schemes reported in [30],

EC: is the embedding capacity,

$|L|$: is the location map length used in [30],

$|\alpha|$: is the 37 bits additional information for a one-pass embedding in the proposed algorithm,

$|C|$: is the length of saved bits of changeable pairs ($20 bits$ for a one-pass embedding in the proposed algorithm),

$|W|$: is the watermark (payload) length (P in the proposed algorithm),

The symbol '$-$' in tables 5 and 6 denotes that the parameter $|L|$ or $|\alpha|$ is not used in the algorithm.

It is clear from Tables 5 and 6 that the proposed method performs well especially in the case of images with large smooth regions. However, its performance is slightly worse in the case of images from the texture category of the database. It should be mentioned that a multi-pass embedding is needed to reach larger embedding capacities. For instance, 2 passes to reach embedding capacity of 131072 *bits* and 4 or 5 passes to reach 262144 *bits*. For Baboon image, 9 passes are used to reach 242864 *bits*.

TABLE 5. Performance comparison of the proposed method and the DER schemes in [30] for $EC = 131072$

Test image	Scheme	*EC*	$\|L\|$	$\|\alpha\|$	$\|C\|$	$\|W\|$	*ER* (*bpp*)	*PSNR*	*SSIM*
Lena	DER scheme in [30]	131072	1530	-	60	129428	0.5	35.76	0.95
	Proposed algorithm	131072	-	74	40	130958	0.5	39.57	0.96
Baboon	DER scheme in [30]	131071	2673	-	121	128205	0.499	31.31	0.93
	Proposed algorithm	131072	-	74	40	131015	0.5	31.61	0.95
Truck	DER scheme in [30]	131072	774	-	30	130196	0.5	37.09	0.96
	Proposed algorithm	131072	-	74	40	130958	0.5	37.26	0.95
APC	DER scheme in [30]	131072	927	-	43	130030	0.5	38.08	0.95
	Proposed algorithm	131072	-	74	40	130958	0.5	38.84	0.95

Table 6. Performance comparison of the proposed method and the DER schemes in [30] for $EC = 262144$

Test image	Scheme	EC	$\|L\|$	$\|\alpha\|$	$\|C\|$	$\|W\|$	ER (bpp)	$PSNR$	$SSIM$
Lena	DER scheme in [30]	262144	26739	-	1981	233280	1.0	31.02	0.71
	DER layer-2 scheme in [30]	262142	26667	-	1971	233360	0.99	31.05	0.76
	Proposed algorithm	262144	-	148	80	261916	1.0	32.52	0.80
Baboon	DER scheme in [30]	262142	72171	-	6829	182998	0.99	28.73	0.71
	DER layer-2 scheme in [30]	262124	72045	-	6817	183118	1.0	28.77	0.71
	Proposed algorithm	242864	-	333	180	261631	0.93	26.81	0.84
Truck	DER scheme in [30]	262144	7020	-	708	254596	1.0	31.13	0.77
	DER layer-2 scheme in [30]	262143	26667	-	6894	254727	0.99	31.21	0.77
	Proposed algorithm	262144	-	185	100	261859	1.0	29.40	0.78
APC	DER scheme in [30]	262144	7893	-	597	253510	1.0	31.60	0.73
	DER layer-2 scheme in [30]	262141	7848	-	592	253557	0.99	31.66	0.74
	Proposed algorithm	262144	-	148	80	261916	1.0	30.53	0.77

5. Conclusions.

In this paper, we have designed a novel algorithm for reversible image watermarking. It is based on the DE method, which is known for its simplicity. The location map considered to be a major drawback of the existing DE-based data hiding techniques is no longer necessary for the proposed algorithm. The main idea behind this very interesting result is the introduction of an appropriate selection of pixel pairs from the smoothest regions of the image using a special case expandability criterion. It has been shown that only 57 *bits* are needed to be embedded in the host image for perfect recovering using the proposed algorithm in the case of a 512×512 8-*bit* gray-scale image. Therefore, the size of the pure payload can considerably be increased by employing the proposed algorithm. Moreover, it has experimentally been shown that the proposed algorithm outperforms the existing DE-based techniques in terms of *PSNR* and *SSIM*. This would make the proposed algorithm a good alternative for efficient reversible image watermarking.

In perspective, the algorithm could be further optimized to process highly textured images, where smooth regions are scarce, which would potentially improve its embedding capability in such cases. Moreover, future work could focus on evaluating the robustness of the proposed method against common image processing operations, such as compression or noise addition, to enhance its practical applicability in real-world scenarios.

Declaration of competing interest.

- The authors have no known competing financial interests that could influence the work reported in this article;
- The authors have no known personal relationships that could influence the work reported in this article.

REFERENCES

[1] P. Niu, F. Wang, and X. Wang, *SVD-UDWT Difference Domain Statistical Image Watermarking Using Vector Alpha Skew Gaussian Distribution*, Circuits Syst Signal Process, vol. 43, pp. 224–263, 2024. https://doi.org/10.1007/s00034-023-02460-w

[2] A. Amiri and B. Kimiaghalam, *Robust Watermarking with PSO and DnCNN*, Signal, Image and Video Processing, (2024). https://doi.org/10.21203/rs.3.rs-3885225/v1

[3] D. Thi-Kien, N. Trong-The, N. Thi-Xuan-Huong, and N. Trinh-Dong, *Recent Information Hiding Techniques in Digital Systems: A Review*, Journal of Information Hiding and Multimedia Signal Processing, vol. 15, no. 1, pp. 10–20, March 2024.

[4] L. Kumar Gupta, A. Singh, A. Kushwaha, and A. Vishwakarma, "Analysis of Image Steganography Techniques for Different Image Format", *International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, Bhilai, India, pp. 1-6, 2021. https://doi.org/10.1109/ICAECT49130.2021.9392492

[5] A. Nandakumar, P. Harmya, N. Jagadeesh, and S. S. Anju, *A secure data hiding scheme based on combined steganography and visual cryptography methods*, Communications in Computer and Information Science, vol. 191, pp. 498-505, 2011. https://doi.org/10.1007/978-3-642-22714-1_51

[6] N. K. Sao, C. T. Luyen, and P. V. At, *Efficient Reversible Data Hiding Using Block Histogram Shifting with Invariant Peak Points*, Journal of Information Hiding and Multimedia Signal Processing, vol. 13, no. 1, pp. 78–97, March 2022.

[7] J. M. Barton, *Method and apparatus for embedding authentication information within digital data*, U.S. Patent 5,646,997, 1997.

[8] C. W. Honsinger, P. W. Jones, M. Rabbani, and J. C. Stoffel, *Lossless recovery of an original image containing embedded data*, U.S. Patent 6,278,791, 2001.

[9] R. Kumar, D. Sharma, A. Dua, and K.-H. Jung, *A Review of Different Prediction Methods for Reversible Data Hiding*, Journal of Information Security and Applications, (2023). https://doi.org/10.1016/j.jisa.2023.103572

[10] Y. Hu, B. Jeon, Z. Lin, and H. Yang, *Analysis and Comparison of Typical Reversible Watermarking Methods*, IWDW 2006, LNCS 4283, pp. 333–347, 2006. https://doi.org/10.1007/11922841_27

[11] O. M. Al-Qershi and K. B. Ee, *An Overview of Reversible Data Hiding Schemes based on Difference Expansion Technique*, ICSECS09, 2009.

[12] R. Caldelli, F. Filippini, and R. Becarelli, *Reversible watermarking techniques: An overview and a classification*, EURASIP Journal on Information Security, vol. 2010, Art. no. 134546, 2010. https://doi.org/10.1155/2010/134546

[13] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, *Lossless generalized-LSB data embedding*, IEEE Transactions on Image Processing, vol. 14, no. 2, pp. 253–266, 2005. https://doi.org/10.1109/TIP.2004.840686

[14] J. Fridrich, M. Goljan, and R. Du, *Invertible authentication*, Proceedings of SPIE, Security watermarking of multimedia contents, San Jose, California, pp. 197–208, 2001. https://doi.org/10.1117/12.435400

[15] Z. Ni, Y. Shi, Q. N. Ansari, and W. Su, *Reversible data hiding*, IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 3, pp. 354–362, 2006. https://doi.org/10.1109/TCSVT.2006.869964

[16] M. Fallahpour and M. H. Sedaaghi, *High-capacity lossless data hiding based on histogram modification*, IEICE Electronics Express, vol. 4, no. 7, pp. 205–210, 2007. https://doi.org/10.1587/elex.4.205

[17] S. K. Lee, Y. H. Suh, and Y. S. Ho, *Reversible image authentication based on watermarking*, Proc. IEEE Int. Conf. Multimedia Expo, pp. 1321–1324, 2006. https://doi.org/10.1109/ICME.2006.262782

[18] 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 Processing, vol. 93, pp. 198–205, 2013. https://doi.org/10.1016/j.sigpro.2012.07.025

[19] X. Qu and H. J. Kim, *Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding*, Signal Processing, vol. 111, pp. 249–260, 2015. https://doi.org/10.1016/j.sigpro.2015.01.002

[20] S. Meikap and B. Jana, *Directional PVO for reversible data hiding scheme with image interpolation*, Multimedia Tools and Applications, vol. 77, pp. 31281–31311, 2018. https://doi.org/10.1007/s11042-018-6203-2

[21] Z. Pan, X. Gao, E. Gao, and G. Fan, *Adaptive complexity for pixel-value-ordering based reversible data hiding*, IEEE Signal Processing Letters, vol. 27, pp. 915–919, 2020. https://doi.org/10.1109/LSP.2020.2996507

[22] Y. Niu and S. Shen, *A novel pixel value ordering reversible data hiding based on dual-image*, Multimedia Tools and Applications, vol. 81, pp. 13751–13771, 2022. https://doi.org/10.1007/s11042-022-12149-y

[23] F. S. Hassan and A. Gutub, *Efficient Image Reversible Data Hiding Technique Based on Interpolation Optimization*, Arabian Journal for Science and Engineering, vol. 46, pp. 8441–8456, 2021. https://doi.org/10.1007/s13369-021-05529-3

[24] J. Tian, *Reversible data embedding using a difference expansion*, IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 8, pp. 890–896, 2003. https://doi.org/10.1109/TCSVT.2003.815962

[25] M. Alattar, *Reversible watermark using the difference expansion of a generalized integer transform*, IEEE Transactions on Image Processing, vol. 13, no. 8, pp. 1147–1156, 2004. https://doi.org/10.1109/TIP.2004.828418

[26] D. M. Thodi and J. J. Rodríguez, *Expansion Embedding Techniques for Reversible Watermarking*, IEEE Transactions on Image Processing, vol. 16, pp. 721–730, 2007. https://doi.org/10.1109/TIP.2006.891046

[27] L. Kamstra and H. Heijmans, *Reversible data embedding into images using wavelet techniques and sorting*, IEEE Transactions on Image Processing, vol. 14, no. 12, pp. 2082–2090, 2005. https://doi.org/10.1109/TIP.2005.859373

[28] H. J. Kim, V. Sachnev, Y. Q. Shi, J. Nam, and H. G. Choo, *A novel difference expansion transform for reversible data embedding*, IEEE Transactions on Information Forensics and Security, vol. 3, no. 3, pp. 456–465, 2008. https://doi.org/10.1109/TIFS.2008.924600

[29] C. C. Lin, S. P. Yang, and N. L. Hsueh, *Lossless Data Hiding Based on Difference Expansion without a Location Map*, IEEE Congress on Image and Signal Processing, pp. 8–12, 2008. https://doi.org/10.1109/CISP.2008.64

[30] S. Gujjunoori and M. Oruganti, *Difference expansion based reversible data embedding and edge detection*, Multimedia Tools and Applications, vol. 78, pp. 25889–25917, 2019. https://doi.org/10.1007/s11042-019-07767-y

[31] V. Sachnev, H. J. Kim, J. Nam, et al., *Reversible Watermarking Algorithm Using Sorting and Prediction*, IEEE Transactions on Circuits and Systems for Video Technology, vol. 19, pp. 989–999, 2009. https://doi.org/10.1109/TCSVT.2009.2020257

[32] W. Wang, *A reversible data hiding algorithm based on bidirectional difference expansion*, Multimedia Tools and Applications, vol. 79, pp. 5965–5988, 2019. https://doi.org/10.1007/s11042-019-08255-z

[33] H. S. El-sayed, S. F. El-Zoghdy, and O. S. Faragallah, *Adaptive Difference Expansion-Based Reversible Data Hiding Scheme for Digital Images*, Arabian Journal for Science and Engineering, pp. 1091–1107, 2016. https://doi.org/10.1007/s13369-015-1956-7