Journal of Information Hiding and Multimedia Signal Processing ©2023 ISSN 2073-4212
Ubiquitous International Volume **14**, Number **1**, March **2023**

Reversible Data Hiding based on dual images adapts to the secret message

Nguyen Kim Sao
Department of Information Technology
University of Transport and Communications
No.3 Cau Giay Street, Lang Thuong ward, Dong Da District, Hanoi, Vietnam
saonkoliver@utc.edu.vn

Tran Xuan Thanh
Department of Information Technology
East Asia University of Technology
Polyco Building, Trinh Van Bo Street, Nam Tu Liem District, Hanoi, Vietnam
thanhtx@eaut.edu.vn

Mai Thanh Hong
Department of Information Technology
Ha Noi University of Industry
No. 298, Cau Dien Street, Bac Tu Liem District, Hanoi, Vietnam
hongmt@fit-haui.edu.vn

Received November 2022; revised February 2023

Abstract. This paper proposes dual images based on reversible data hiding adapted secret bits. It calculates the frequencies of 5-bit elements of secret bits and builds the rank table for them to determine which parameters will be plus to the cover pixels. The rank table help to reduce the distortion of two stego images when the payload is less than capacity. The cover image is divided into blocks, and 8 bits will be embedded in each block. The experimental results and analysis show that the proposed method has a high embedding capacity with good image quality at both stego images.
Keywords: data hiding, dual images, frequency, RDH, dual images RDH

1. Introduction.

Reversible data hiding (RDH) is a technique that embeds a secret message into a cover image. Unlike traditional data hiding which only extracts the secret bits at the decode's side, RDH can restore the cover image beside extract the secret message.

The first RDH was proposed by Barton in 1997 [1], since there, many techniques are used in reversible data hiding, like as Modulo transform, lossless compression, quantization, RDH in JPEG images, Difference Expansion, Prediction Error Expansion, PVO.

If considering the number of stego images, RDH can classify into two kinds that is one stego and more than one stego image. Since 1997, almost proposed methods regenerate only one stego image, the secret message is embedded in a cover image and generates a stego image, from this stego image, the secret data and cover image could receive. Since 2007, C.C. Chang et al. [3] proposed an RDH method that gives two stego images, this kind has received significant attention from many researchers. This is a scheme that defines the rule to embed secret data in a cover image but generates two stego images.

To extract secret data and restore the cover images, two stego images need to be used. In the first method, Chang et al. extended the EMD (Exploiting modification direction) algorithm by using 5×5 modulo matrix, two stego images are obtained by embedding data into pixels according to the index values of two corresponding diagonals in the matrix. The method proposed by C.C. Chang et al. in [2] is the improvement of [3]. In this, the authors change direction from diagonal to horizontal and vertical to improve the stego image quality. expanded matrix to 9×9 size to gain higher capacity. T.C. Lu et al. [9] used the folding strategy to decrease the values of hidden values.

H. Yao et al. [13] gave a prediction-error shift strategy. In [13], the embedding procedure is divided into two phases. The pixels of the host image are classified into two kinds, the plank pixels and padding pixels based on rhombus pattern prediction. In the first phase, the secret message will be embedded in padding pixels and the blank pixel is used for prediction, in the second phase, the process is reversed. T.C. Lu et al. [10] used the center folding strategy to reduce the values of sets of secret bits. This strategy aims to reduce pixel variability. H. Yao et al. [14] put more than one bit to the group of k bit in [10] by adding this bit into k-bits in decimal. S. Shastri et al. [12] is an improvement of [10] and [14]. [12] also use a center folding strategy and in their scheme, it produces the passwords that are a location map to determine whether a group of secret bits in decimal is odd or even. These passwords will be sent to the receiver for extraction.

X. Chen and W. Guo [5] use a rule table for each group of embedded k-bits. Each k-bits will be embedded in pair of pixels based on the rule table. In this scheme, each pixel is changed maximum is 1. Another scheme of X. Chen about dual-image RDH is [6], in this proposal X. Chen and C. Hong establishes exploiting modification direction (EMD) for the embedding scheme, the algorithm will embed 6 bits into a couple of pixels. J. Hsiao et al. [7] propose a method based on EMD called the AEMD scheme that gives a weight of the second pixel in an embedded pixels-pair. This weight depends on the number of groups of bits embedded in a couple of pixels. C.C.Chang et al. [4] use a mark based on area, what is called a sunflower, to determine which case is used for changing the pixels. Y Niu and S Shen [11] divided the host image into blocks and sort the pixels ascending and apply the pixel value ordering method to embed the secret data.

This paper proposes an improvement in the method of X. Chen and W. Guo [5], instead of embedding 4 or 5 bits for a pair of pixels, the proposed method embeds 5 pixels for each block sized 2×1. To maintain the stego image quality, the rank table is established to determine which parameters will use for embedding. The higher frequency, the less change in the cover pixels. With this improvement, our method achieves a higher capacity and has the same or less insignificant image quality than [5].

The rest of this paper is organized as follows. Section 2 reviews the method of X.Chen and W. Guo that is the related method to the proposed method. Section 3 introduces the proposed method. Experimental results are presented in Section 4. Finally, the conclusion is made at the end of the paper.

2. **Review the scheme of X. Chen and W. Guo .**

X. Chen and W.Guo [5] improves the method of C.F. Lee and Y.L. Huang [8]. First, the authors use 3 blocks to create a mask for a pair of embedded pixels. With a pair, the embedded bits will use a suitable mask. With a pair of cover pixels (P_i, P_{i+1}), [5] will embed 5 or 4 secret bits to get a couple of stego pairs $(M_i, M_{i+1}), (A_i, A_{i+1})$. The rule to embed follow the Table 1.

v	Secret Bits	(M_i, M_{i+1})	MM	(A_i, A_{i+1})	AM	(d_i, d_{i+1})	(P_i, P_{i+1})
0	00000	(P_i, P_{i+1})	0	(P_i, P_{i+1})	0	(0,0)	(M_i, M_{i+1})
1	00001			$(P_i, P_{i+1}+1)$	1	(0,1)	
2	00010			(P_i+1, P_{i+1})	2	(1,0)	
3	00011			$(P_i, P_{i+1}-1)$	3	(0,-1)	
4	00100			(P_i-1, P_{i+1})	4	(-1,0)	
5	00101			$(P_i+1, P_{i+1}+1)$	5	(1,1)	
6	00110			$(P_i+1, P_{i+1}-1)$	6	(1,-1)	
7	00111			$(P_i-1, P_{i+1}-1)$	7	(-1,-1)	
8	01000			$(P_i-1, P_{i+1}+1)$	8	(-1,1)	
9	01001	$(P_i, P_{i+1}+1)$	1	$(P_i, P_{i+1}-1)$	3	(0,2)	$(M_i, M_{i+1}-1)$
10	01010			$(P_i+1, P_{i+1}-1)$	6	(-1,2)	
11	01011			$(P_i-1, P_{i+1}-1)$	7	(1,2)	
12	01100	(P_i+1, P_{i+1})	2	(P_i-1, P_{i+1})	4	(2,0)	(M_i-1, M_{i+1})
13	01101			$(P_i-1, P_{i+1}-1)$	7	(2,1)	
14	01110			$(P_i-1, P_{i+1}+1)$	8	(2,-1)	
15	01111	$(P_i, P_{i+1}-1)$	3	$(P_i-1, P_{i+1}+1)$	1	(1,-2)	$(M_i, M_{i+1}+1)$
16	10000			$(P_i+1, P_{i+1}+1)$	5	(-1,-2)	
17	10001			$(P_i, P_{i+1}+1)$	8	(0,-2)	
18	1001	(P_i-1, P_{i+1})	4	(P_i+1, P_{i+1})	2	(-2,0)	(M_i+1, M_{i+1})
19	1010			$(P_i+1, P_{i+1}-1)$	5	(-2,1)	
20	1011			$(P_i+1, P_{i+1}+1)$	6	(-2,-1)	
21	1100	$(P_i+1, P_{i+1}+1)$	5	$(P_i-1, P_{i+1}-1)$	7	(2,2)	$(M_i-1, M_{i+1}-1)$
22	1101	$(P_i+1, P_{i+1}-1)$	6	$(P_i-1, P_{i+1}+1)$	8	(2,-2)	$(M_i-1, M_{i+1}+1)$
23	1110	$(P_i-1, P_{i+1}-1)$	7	$(P_i+1, P_{i+1}+1)$	5	(-2,-2)	$(M_i+1, M_{i+1}+1)$
24	1111	$(P_i-1, P_{i+1}+1)$	8	$(P_i+1, P_{i+1}-1)$	6	(-2,2)	$(M_i+1, M_{i+1}-1)$

TABLE 1. The embedding and extracting rules of the x.Chen and W.Guo's method [5] (MM: Major Mark; AM: Auxiliary Mark)

In the embedded algorithm, they take 5 bits from the array of embedded bits. Convert it to decimal v, if $v \geq 17$ then 5-bits will be embedded, otherwise, only takes 4 bits for embedding.

At the extracting algorithm, the host pair is restored of pixels and secret bits are extracted as follows:

First, the difference between the two pair are calculated:

$$(d_i, d_{i+1}) = (M_i - A_i, M_{i+1} - A_{i+1})$$

From $(d_i, d_{i+1}), (M_i, M_{i+1})$ and based on Table 1 to get secret bits (second column) and host pixels (P_i, P_{i+1}) (8th column).

3. **Proposed methods.**

3.1. **The rule of embedding for a couple of pixel .**

Because the proposed method is adaptive with the secret bits B to reduce distortion of stego image, the rank of secret 5-bit elements is used to specify which element has the highest frequency will embed with the lowest transform. The rank of an element depends on its frequency in the array of secret bits. Corresponding with rank, there are α_i, α_{i+1}, β_i, β_{i+1} that present the transformation of the cover block to two stego blocks. Table 2 shows the transformed parameter $\alpha_i, \alpha_{i+1}, \beta_i, \beta_{i+1}$, the difference between two pixels of two stego blocks d_i, d_{i+1} corespond with rank at the first column.

Rank (r)	α_i, α_{i+1}	β_i, β_{i+1}	d_i, d_{i+1}	Rank	α_i, α_{i+1}	β_i, β_{i+1}	d_i, d_{i+1}
0	0, 0	0, 0	0,0	16	0, 1	1, -1	-1,2
1	0, -1	0, 0	0,-1	17	0, -1	-1, 1	1,-2
2	-1, 0	0, 0	-1,0	18	1, 1	0, -1	1,2
3	0, 0	0, -1	0,1	19	1, -1	-1, 0	2,-1
4	0, 0	-1, 0	1,0	20	1, 0	-1, -1	2,1
5	-1, 0	1, 0	-2,0	21	-1, -1	1, 1	-2,-2
6	-1, 0	0, 1	-1,-1	22	-1, 1	1, -1	-2,2
7	0, 1	1, 0	-1,1	23	1, -1	-1, 1	2,-2
8	0, -1	0, 1	0,-2	24	1, 1	-1, -1	2,2
9	0, 1	0, -1	0,2	25	2, 0	-1, 0	3,0
10	0, -1	-1, 0	1,-1	26	0, 1	0, -2	0,3
11	1, 0	0, -1	1,1	27	-2, 0	1, 0	-3,0
12	1, 0	-1, 0	2,0	28	0, -1	0, 2	0,-3
13	-1, -1	1, 0	-2,-1	29	2, 1	-1, 0	3,1
14	-1, 0	1, -1	-2,1	30	0, 1	-1, -2	1,3
15	0, -1	1, 1	-1,-2	31	-1, 1	0, -2	-1,3

TABLE 2. Rank Table

3.2. Embedding algorithm in a block.

Input: A block with size 1×2 (P_i, P_{i+1}), aray of 5 secret bit $b_1, \ldots, b_5$, the rank r of array of secret bits.

Output: Two stego-blocks $(M_i, M_{i+1}), (A_i, A_{i+1})$

Case 1: If the block contains one of the values $\{0, 1, 254, 255\}$, ignore this block, two stego blocks are the same original block. if not, change to case 2.

Case 2: From the rank r of an array of $b_1, b_2 \ldots b_5$ bits (see Table 2), embedding 5 secret bits $b_1, \ldots, b_5$ as follows:

$$\begin{aligned} M_i &= P_i + \alpha_i^r, \\ M_{i+1} &= P_{i+1} + \alpha_{i+1}^r, \\ A_i &= P_i + \beta_i^r, \\ A_{i+1} &= P_{i+1} + \beta_{i+1}^r, \end{aligned}$$

which $\alpha_i^r, \alpha_{i+1}^r, \beta_i^r, \beta_{i+1}^r$ are the parameters at the line r, correspone with the rank of set of embedded bits.

3.3. Extracting and restoring algorithm in a block.

Input: Two stego-blocks (M_i, M_{i+1}), (A_i, A_{i+1}), the rank table of array of secret bits

Output: Original block (P_i, P_{i+1}), 5 secret bits $b_1, b_2, \ldots, b_5$

First of all, calculate

$$\begin{aligned} d_i &= M_i - A_i, \\ d_{i+1} &= M_{i+1} - A_{i+1}. \end{aligned}$$

Case 1: If the stego block contains one of the values $\{0, 1, 254, 255\}$, and $d_i = 0; d_{i+1} = 0$, there is not any secret bits, the original block is the same stego block. If not, switch to case 2.

Case 2: From d_i, d_{i+1}, locate the d_i, d_{i+1} column in Table 2, get the rank r of bit array, therefrom, getting 5 secret bits $b_1, b_2, \dots, b_5$ correspond with the rank.

Restoring the original block as below:

$$P_i = M_i - \alpha_i^r,$$
$$P_{i+1} = M_{i+1} - \alpha_{i+1}^r,$$

which $\alpha_i^r, \alpha_{i+1}^r$ are the parameters at the line r.

So that, we get 5 secret bits $b_1, b_2, \dots, b_5$ and the cover block P_i, P_{i+1}

3.4. **Underflow/overflow processing.**

The parameters in $[-2; 2]$ will be plus into pixels. To avoid underflow/ overflow, the block has values in $[0, 1, 254, 255]$ will be ignored.

To distinguish the stego blocks which have values $0, 1, 254, 255$ contain the secret bits or not, compare two stego blocks, if they are the same, they do not contain the secret bits, if not, they have secret bits.

For example, two stego blocks are $(1, 15)$ and $(1, 15)$, so that $d_1 = 0; d_2 = 0$, we know there are not any bits embedded in this block, but if two stego blocks are $(1, 15)$ and $(3, 15)$, we have $d_1 = -2, d_2 = -0$, see the rank table, $r = 27$, from the rank of a set of 5 bits, we will get the secret bits and the stego block is $(3, 15)$.

3.5. **Examples for illustrating embedding and extracting algorithm in a block.**

Embedding:

Supposing an original block is (230, 235), and secret bits is 10011 and the rank of it is 9.

from the rank (is 9), we know that $\alpha_i^r = 0$, $\alpha_{i+1}^r = 1$,

$$\beta_i^r = 0$$

, $\beta_{i+1}^r = -1$. So that: $M - i = 230 + 0 = 230$, $M_{i+1} = 235 + 1 = 236$, $A_i = 230 + 0 = 230$, $A_{i+1} = 235 - 1 = 234$, then two stego blocks are (230, 236) and (230, 234).

Extracting and restoring:

With two stego blocks (230, 236) and (230, 234). Firstly, calculating $d_i^r = 230 - 230 = 0$, $d_{i+1}^r = 236 - 234 = 2$. So that, from Table 2, we know the rank is 9. From rank (is 9), to get the array of secret bits is 10011 (the rank of the array of secret bits will count before embedding and saving in the stego image). And Finding the $\alpha_i^r = 0$, $\alpha_{i+1}^r = 1$.

Restoring the original block: $P_i = 230 - 0 = 230$, $P_{i+1} = 236 - 1 = 235$, so that the results are (230, 235).

3.6. **Defining the auxiliary information .**

To extract the secret and restore the stego image, there is information needs to be used.

The first is the rank table, we need $5 \times 32 = 160$ bits to save it. The rank will be saved in order of the descending frequency of 5-bit elements, the element has max frequency at the front and the min at the last.

The second information is the position of the block that starts embedding the extraLSB. This information is used for restoring the stego image matrix before embedding auxiliary information in it by LSB.

	Content	Size
1	The Rank Table of 5-bits number	$5 \times 32 = 160$
2	The position of the block which embed the extraLSB	16

TABLE 3. Auxiliary information

3.7. **Data embedding procedure.**

Here, the embedding process is described in detail.

Input: A 8-bit grayscale host image I and the data bits B.

Output: Two stego image I' and I".

Step 1: Building the rank table

Partitioning the data bits B into the array of the 5-bit elements. Calculating the frequency of 5-bit element in the array and sorting it in descending order. To extract, this order will embed in stego. The order will save as 5-bit element by element.

Exampale B is 00101110101011000101110101101011010, partitioning it in to array: 00101/11010/10110/00101/11010/11010/11010, call $F(x)$ is frequency of x, so $F[00101] = F[5] = 2, F[11010] = F[26] = 4, F[10110] = F[22] = 1, F(i) = 0$ $(i = 0, \dots, 31$ *and* $i \neq 5, 22, 26)$, sort the $F(x)$ in descending order, we get $\{F(26), F(5), F(22), F(i)$ $(i = 0, \dots, 31$ *and* $i \neq 5, 22, 26)\}$, the rank are $R[26] = 0, R[5] = 1, R[22] = 2, R(i) = k$ $(i = 0, \dots, 31$ *and* $i \neq 5, 22, 26, k = 3, \dots, 31)$. The order will be 110100010110110000000000 100010000011001000011000011 101000010010101001011011O 001101011100111110000100O1 100101001110100101011011111 000110011101111100111011111011111

Step 2: Divide I into non-overlapped 1×2-sized blocks. Visiting block by block and embedding data B in block according to subsection 3.2.

Step 3: Getting the position of the next block of the block that ends embedding B. Taking 176 LSB of 176 first pixels of the image (call extraLSB), continue embedding these 176 bits into the image.

Step 4: Embedding the auxiliary information into 176 first pixels of the image by LSB method to get the stego image.

3.8. **Data extracting procedure.**

Here, the data will be extracted and the host image will be restored as follow:

Input: A 8-bit grayscale stego image I'

Output: Data bits B and host image I.

Step1. Take 176 LSB bits from 176 first pixels of the stego image I' to get the rank of 5-bits elements and the position of the block which embed the extraLSB.

Step 2. Divide I' into 1×2-sized blocks as the same in the data embedding procedure.

Step 3. From the position of the block which embeds the extraLSB, Extracting 176 embedded bits and restoring host blocks according to the algorithm as in subsection 3.3. Inserting extracted bits into 176 LSB bits of the 176 first pixels of I'.

Step 4. Continuing Extracting data bits B and restoring original blocks beginning from the first block by using the extracting algorithm as in subsection 3.3.

At last, data bits B are extracted and the original image I is restored.

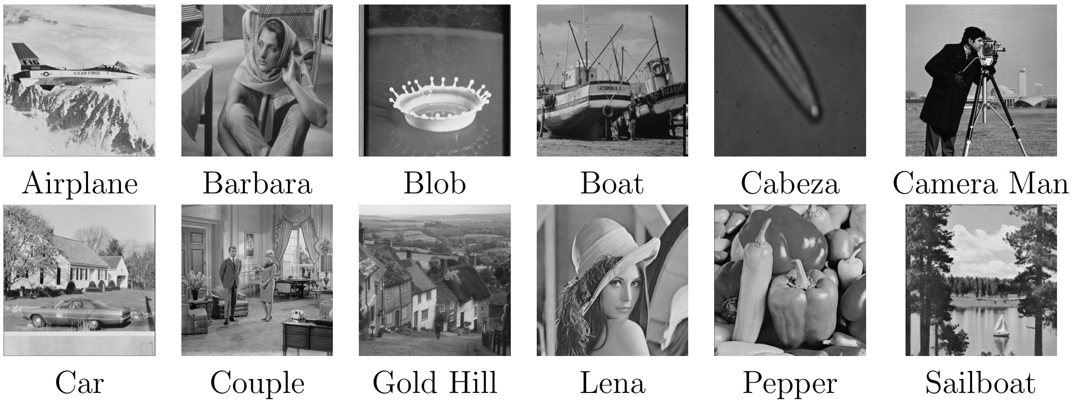


Figure 1. Experimental images

4. **Experimental results.**

TT	Images	[11](2 × 2)	[11](3 × 1)	[5]	[4]	Proposed
1	Airplane	0.563	0.747	1.218	1.187	1.250
2	Barbara	0.562	0.747	1.217	1.187	1.250
3	Blob	0.563	0.747	1.218	1.188	1.250
4	Boat	0.562	0.747	1.217	1.186	1.250
5	Cabeza	0.562	0.747	1.217	1.188	1.250
6	Camera man	0.561	0.746	1.216	1.183	1.248
7	Car	0.563	0.747	1.218	1.185	1.250
8	Couple	0.561	0.745	1.215	1.182	1.248
9	Gold hill	0.563	0.747	1.218	1.188	1.250
10	Lena	0.563	0.747	1.218	1.187	1.250
11	Pepper	0.562	0.747	1.217	1.185	1.250
12	Sailboat	0.563	0.747	1.218	1.187	1.250
	Average	0.562	0.747	1.217	1.186	1.250

Table 4. Comparison results in terms of embedding capacity

To illustrate the efficiency of the proposed method and results of theoretical analysis, we perform experiments on 15 standard grayscale images of size 512 × 512 selected as shown in Figure 1 for comparing 3 recent methods: X.Chen and W. Guo [5], Y.Niu and S. Shen [11], C.C.Chang et al.[4]. All the methods were implemented through Matlab R2019a.

We use two terms to compare the efficiency of each method. Bits per pixel (bpp) for capacity and PSNR (peak signal of noise ratio) for image quality. PSNR, is the difference between the cover image and stego images, is calculated in the formula as follows:

$$PSNR = 10 \times log_{10} \left[\frac{255^2}{\frac{1}{M \times N} \times \sum_{i=1}^{M} \sum_{j=1}^{N} (x'_{ij} - x_{ij})^2} \right] (dB),$$

where $M \times N$ is the size of the image and x_{ij}, x'_{ij} are pixels of the stego image and cover image, respectively. The higher PSNR, the better the image quality. The capacity ratios can calculate by:

$$\beta = \frac{C}{n \times M \times N}$$

TT	Images	[11]		[5]		[4]		Proposed	
		I1	I2	I1	I2	I1	I2	I1	I2
1	Airplane	58.13	58.17	59.10	56.76	57.26	53.38	57.74	57.41
2	Barbara	58.38	58.45	59.10	56.76	57.12	53.45	57.74	57.41
3	Blob	58.29	58.35	59.10	56.76	57.33	53.36	57.74	57.41
4	Boat	58.56	58.49	59.10	56.76	57.28	53.40	57.74	57.41
5	Cabeza	57.88	57.84	59.10	56.76	57.35	53.41	57.74	57.41
6	Camera man	57.64	57.73	59.10	56.76	57.20	53.45	57.74	57.41
7	Car	58.35	58.56	59.10	56.76	57.24	53.44	57.74	57.41
8	Couple	58.84	58.74	59.10	56.76	57.25	53.42	57.74	57.41
9	Gold hill	58.79	58.68	59.10	56.76	57.37	53.37	57.74	57.41
10	Lena	58.56	58.49	59.10	56.76	57.29	53.45	57.74	57.41
11	Pepper	58.73	58.80	59.10	56.76	57.29	53.42	57.74	57.41
12	Sailboat	58.77	58.79	59.10	56.76	57.26	53.39	57.74	57.41
	Average	58.41	58.43	59.10	56.76	57.27	53.41	57.74	57.41
		58.49		57.93		55.34		57.57	

TABLE 5. Comparison results in terms of PSNR with payload are 100000 bits

TT	Images	[11]		[5]		[4]		Proposed	
		I1	I2	I1	I2	I1	I2	I1	I2
1	Airplane	54.62	54.62	55.09	52.77	53.30	49.40	53.73	53.39
2	Barbara	54.90	54.91	55.09	52.77	53.19	49.46	53.73	53.39
3	Blob	54.53	54.57	55.09	52.77	53.29	49.42	53.73	53.39
4	Boat	55.30	55.32	55.09	52.77	53.24	49.42	53.73	53.39
5	Cabeza	53.98	53.94	55.09	52.77	53.32	49.43	53.73	53.39
6	Camera man	54.13	54.15	55.09	52.77	53.24	49.42	53.73	53.39
7	Car	54.83	54.93	55.09	52.77	53.27	49.43	53.73	53.39
8	Couple	55.53	55.47	55.09	52.77	53.33	49.41	53.73	53.39
9	Gold hill	55.54	55.49	55.09	52.77	53.31	49.41	53.73	53.39
10	Lena	55.07	55.03	55.09	52.77	53.35	49.43	53.73	53.39
11	Pepper	55.37	55.41	55.09	52.77	53.20	49.46	53.73	53.39
12	Sailboat	55.60	55.64	55.09	52.77	53.26	49.43	53.73	53.39
	Average	54.95	54.96	55.09	52.77	53.27	49.43	53.73	53.39
		54.95		53.93		51.35		53.56	

TABLE 6. Comparison results in terms of PSNR with payload are 250000 bits

β is the ratio capacity, C is the maximum number of bits that can embed in the image, $M \times N$ is the size of the original image, n is the number of stego images, in this paper $n = 2$. A larger β indicates a larger embedding capacity.

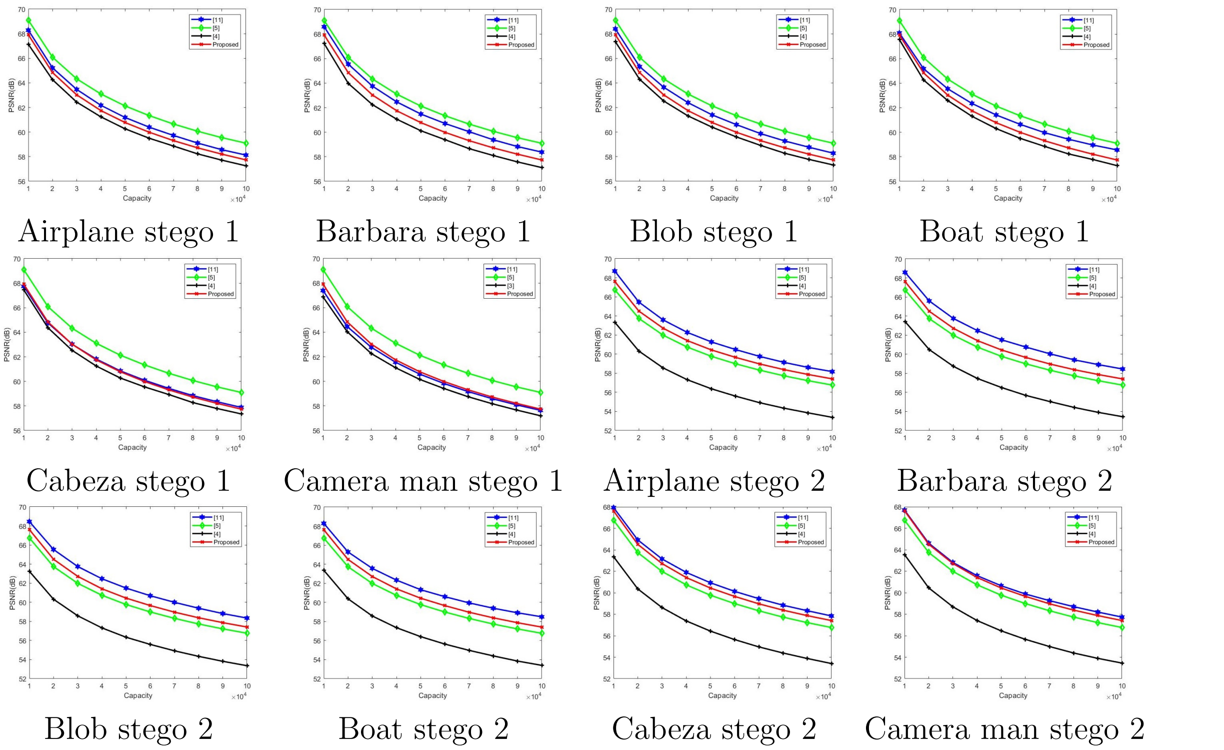
Table 4 presents the comparison of the embedding capacity of methods, the capacity of the proposed method can calculate easily because embedding in every pixel is the same.

By the Table 4, the β of these methods respectively are: [11] with the 2×2 blocks is 0.56, [11] with the 3×1 blocks is 0.75, [5] is 1.22 [4] is 1.19 and the capacity of the proposed method is 1.25, the capacity of the proposed method is highest than that of all.

Table 5 and Table 6 represent the comparison of image quality when the payload is 100000 and 250000 random bits. I1, I2 represent the PSNR of the first and second stego images quality, respectively, the last row is an average of two averages of I1, I2 of each

TT	Images	[11] (2×2) 290000		[11] (3×1) 380000		[5] 630000		[4] 610000		Proposed 640000	
		I1	I2	I1	I2	I1	I2	I1	I2	I1	I2
1	Airplane	54.62	54.62	53.70	53.69	51.31	48.98	49.58	45.72	49.64	49.29
2	Barbara	54.90	54.91	54.07	54.08	51.31	48.98	49.54	45.73	49.64	49.29
3	Blob	54.53	54.57	53.65	53.62	51.31	48.98	49.63	45.70	49.64	49.29
4	Boat	55.30	55.32	54.35	54.37	51.31	48.99	49.58	45.73	49.64	49.29
5	Cabeza	53.98	53.94	53.08	53.12	51.31	48.98	49.61	45.73	49.64	49.29
6	Camera man	54.13	54.15	53.33	53.36	51.32	48.99	49.56	45.73	49.64	49.29
7	Car	54.83	54.93	54.05	54.03	51.31	48.98	49.59	45.74	49.64	49.29
8	Couple	55.53	55.47	54.53	54.50	51.32	48.99	49.61	45.75	49.64	49.29
9	Gold hill	55.54	55.49	54.63	54.54	51.31	48.98	49.59	45.71	49.64	49.29
10	Lena	55.07	55.03	53.97	53.99	51.31	48.98	49.58	45.73	49.64	49.29
11	Pepper	55.37	55.41	54.43	54.43	51.31	48.98	49.57	45.73	49.64	49.29
12	Sailboat	55.60	55.64	54.66	54.67	51.31	48.98	49.58	45.72	49.64	49.29
	Average	54.95	54.96	54.04	54.03	51.31	48.99	49.59	45.73	49.64	49.29
		55.03		54.11		50.15		47.66		49.46	

TABLE 7. Comparison results in terms of PSNR with payload are the maximum capacity



Airplane stego 1 Barbara stego 1 Blob stego 1 Boat stego 1

Cabeza stego 1 Camera man stego 1 Airplane stego 2 Barbara stego 2

Blob stego 2 Boat stego 2 Cabeza stego 2 Camera man stego 2

FIGURE 2. Comparison in terms of PSNR with different embedding payload

method. The PSNR of Y.Niu et al.'s method [11] is higher than that of the proposed method but the capacity of that method is only approximately haft of the capacity of the proposed method. The method of Y. Niu et al. [11] is a PVO method, this method has a character that is high-quality images but low capacity. The method of X. Chen et al. [5] have a PSNR of the first stego image higher than that of the proposed method but the PSNR of the second stego image is lower than that of the proposed method so that the average of I1 and I2 are the same in both methods. The capacity and PSNR of C.C. Chang et al.'s method [4] is lower than that of the proposed method.

Table 7 represents the PSNR of two stego images when embedding the numbers of bits approximate the maximum capacity which is shown in the first row. Although the average PSNR of stego images based on Y.Niu et al.'s method is higher at 5.57 dB than that of the proposed method, the capacity is less than 350000 bits than the proposed method. The PSNR of the proposed method is less than 0.69 dB than that of X.Chen et al.'s method and higher 1,8dB than that of the method of C.C. Chang et al.

In Fig 2, with the image quality of the first stego image, the proposed method (red line) is in the third order, but with the second, the proposed method is in the second order, only less than [11] which is a method base on PVO. According to Fig 2, we can see that the proposed method has good image quality at both stego images.

5. **Conclusion .**

This paper presents a dual-images reversible data hiding method that has a high capacity by embedding 6 bits in each block sized 2×1. To low stego image distortion by using the frequency of the 5-bits elements to rank the rule to embed. Experimental results indicate that the proposed method can achieve 1.25 bpp embedding capacity and the stego image qualities gain 49.46dB for both stego images when embedding maximum capacity.

REFERENCES

[1] J. M. Barton, "Method and apparatus for embedding authentication information within digital data," Jul. 8 1997, uS Patent 5,646,997.

[2] C.-C. Chang, Y.-C. Chou, and T. D. Kieu, "Information hiding in dual images with reversibility," in *2009 Third International Conference on Multimedia and Ubiquitous Engineering.* IEEE, 2009, pp. 145–152.

[3] C.-C. Chang, T. D. Kieu, and Y.-C. Chou, "Reversible data hiding scheme using two steganographic images," in *TENCON 2007-2007 IEEE Region 10 Conference.* IEEE, 2007, pp. 1–4.

[4] C.-C. Chang, G.-D. Su, C.-C. Lin, and Y.-H. Li, "Position-aware guided hiding data scheme with reversibility and adaptivity for dual images," *Symmetry*, vol. 14, no. 3, p. 509, 2022.

[5] X. Chen and W. Guo, "Reversible data hiding scheme based on fully exploiting the orientation combinations of dual stego-images." *Int. J. Netw. Secur.*, vol. 22, no. 1, pp. 126–135, 2020.

[6] X. Chen and C. Hong, "An efficient dual-image reversible data hiding scheme based on exploiting modification direction," *Journal of Information Security and Applications*, vol. 58, p. 102702, 2021.

[7] J.-Y. Hsiao, A.-C. Pan, and P.-Y. Chen, "An adaptive reversible data hiding scheme using dual stego-image," *International Journal of Computers and Applications*, vol. 43, no. 3, pp. 282–291, 2021.

[8] C.-F. Lee and Y.-L. Huang, "Reversible data hiding scheme based on dual stegano-images using orientation combinations," *Telecommunication Systems*, vol. 52, no. 4, pp. 2237–2247, 2013.

[9] T.-C. Lu, C.-Y. Tseng, and J.-H. Wu, "Dual imaging-based reversible hiding technique using lsb matching," *Signal Processing*, vol. 108, pp. 77–89, 2015.

[10] T.-C. Lu, J.-H. Wu, and C.-C. Huang, "Dual-image-based reversible data hiding method using center folding strategy," *Signal Processing*, vol. 115, pp. 195–213, 2015.

[11] Y. Niu and S. Shen, "A novel pixel value ordering reversible data hiding based on dual-image," *Multimedia Tools and Applications*, vol. 81, no. 10, pp. 13 751–13 771, 2022.

[12] S. Shastri and V. Thanikaiselvan, "Dual image reversible data hiding using trinary assignment and centre folding strategy with low distortion," *Journal of Visual Communication and Image Representation*, vol. 61, pp. 130–140, 2019.

[13] H. Yao, F. Mao, Z. Tang, and C. Qin, "High-fidelity dual-image reversible data hiding via prediction-error shift," *Signal Processing*, vol. 170, p. 107447, 2020.

[14] H. Yao, C. Qin, Z. Tang, and Y. Tian, "Improved dual-image reversible data hiding method using the selection strategy of shiftable pixels' coordinates with minimum distortion," *Signal Processing*, vol. 135, pp. 26–35, 2017.