

Reversible Data Hiding based on Improved Reduced Difference Expansion and Multi-layer Recursive Embedding

Chin-Feng Lee ^a, Jau-Ji Shen ^{b*}, Yu-Hua Lai ^b

^a Dept. of Information Management, Chaoyang University of Technology, Taichung, Taiwan

^b Dept. of Management Information Systems, National Chung Hsing University, Taichung, Taiwan

*Corresponding Author: jjshen@nchu.edu.tw

Abstract

This paper proposed a reversible multiple layer data hiding to improve the method proposed by Arham *et al.* using the relevance of pixels to modify the block size and change the scan mode to enhance image embedded capacity. The experimental results shows that the smaller the block, the lower the amount of hidden information, the larger the block, and the higher the hidden amount. This paper suggests appropriate block size for hiding according to the purpose of use, so as to achieve the best performance.

Keywords: Reversible Data Hiding; Difference expansion; Multi-layer embedding.

1. Introduction

High-speed communication networks enable simple and fast online information sharing modes at high data rates. However, the channel for data sharing is not secure. Data hiding technology gives a good way to protect data as it gives a channel to hide information in a host (cover) media without destroying its original value. Reversible data hiding (RDH) in images attracts more and more researchers' attentions. RDH in images aims at embedding imperceptible confidential data in images and allows recipients to extract hidden information and completely recover images. As a result, RDH is widely utilized in the areas of medical, military and law fields where the restored images containing sensitive information can further be used in other important aspects.

Many RDH methods have been proposed so far. The concept of differential expansion was first proposed by Tian in 2003⁽¹⁾ and the differential expansion (DE) RDH method uses the nature of the similarity between adjacent pixels in

an image, selects every two adjacent pixels as a group, calculates the difference value and the mean value, and when hiding confidential information, it can be divided into two kinds situation, expandable or changeable. In order to discriminate between the two situations, the extract information is recorded using a location map. However, the location map will reduce the pure payload of camouflage images. Many scholars later proposed different methods of improvement.

Alatter *et al.*⁽²⁾ calculated the prediction error value of an image pixel by using a weighted average of four adjacent pixels. Liu *et al.*⁽³⁾ designed a function that reduces the difference expansion between the to-be-embedded adjacent pixels such that the image can alleviate the problem of large distortion caused by pixel difference expansion. In 2009, Hu *et al.*⁽⁴⁾ proposed the Improved Reduced Difference Expansion (IRDE) method and they changed the embedding step by using an improved logarithm transformation function to reduce the distortion by difference expansion further more.

In 2010, Wang *et al.* proposed an efficient reversible generalized integer transform hiding method⁽⁵⁾. This method improves the methods of Tian scholars. This method takes n pixels as a unit of embedded messages, each pixel unit can embed secret information of $(n-1)$ bits, and a threshold is predetermined. This threshold is used to adjust image quality and embedding capacity. If the value of threshold is large, the amount of hiding is larger, but the image quality is lower. On the other hand, if the value of threshold is small, the image quality is high, but the amount of hidden information is small. Compared with Tian's DE method, the method proposed by Wang *et al.* can improve image quality.

In 2016, Qiu *et al.* proposed a self-adjusting method⁽⁶⁾ which improves the reversible generalized integer transform hiding method. A group of n pixels is used as an embedded unit, and each pixel unit can embed $m \times (n-1)$ bit secret information, and according to the smoothness of the block,

the self-adjusted adjustment adjusts how much information is hidden, thereby improving the quality of the camouflaged image. And this method has a higher embedded capacity than that of Wang et al's method.

In 2017, Arham *et al.*⁽⁷⁾ combined the methods of quad_DE and IRDE, to improve image quality and embedding capacity. Using the recursive embedding method⁽²⁾ proposed by Alattar can alleviate the problem of large distortion caused by embedded multi-layer images.

This paper will further improve recursive embedding method by altering the image size and the scanning strategy to enhance embedding capacity and image quality.

2. Related Works

The basic idea of the improved reduced difference expansion (IRDE) hiding method is proposed by Hu et al.⁽⁴⁾. The IRDE method calculates the difference value d and the mean value μ between two neighboring pixels x and y . That is $d = x - y$, $\mu = (x + y)/2$, $x = \mu + d/2 + \text{LSB}(d)$, and $y = \mu - d/2$. Here divide(/) is integer divide; for example, x/y means $\text{floor}(x/y)$. For example, the difference value and the mean value between a pair of pixels (110, 95) are $d=110-95=15$ and $\mu = (110+95)/2=102$, respectively. In this manner, $x = 102+7+1=110$ and $y=102-7=95$.

Before data embedding, IRDE method separates the difference value between the pair of pixels x and y as following two classes as Equation (1) such that the difference expansion of IRDE is smaller when the original difference falls within the interval $3 \times 2^{n-1} \leq d \leq 4 \times 2^{n-1} - 1$ with the location map information recorded as $LM = 1$; otherwise, $LM = 0$ when the original difference is within the interval $2 \times 2^{n-1} \leq d \leq 3 \times 2^{n-1} - 1$ where $n = \lceil \log_2 d \rceil$ that present how many number bits of secret data can be hidden in the difference values.

$$d' = \begin{cases} d - 2^{\lceil \log_2 d \rceil - 1}, & \text{if } 2 \times 2^{n-1} \leq d \leq 3 \times 2^{n-1} - 1 \\ d - 2^{\lceil \log_2 d \rceil}, & \text{if } 3 \times 2^{n-1} \leq d \leq 4 \times 2^{n-1} - 1. \end{cases} \quad (1)$$

Two steps of the embedding procedure are as follows. The first step: $d'' = d' \times 2 + \text{one secret bit } (b)$. The second step: $x' = \mu + d'' / 2 + \text{LSB}(d'')$ and $y' = \mu - d'' / 2$.

In the extraction procedure, the location map is used to extract the hidden data. First step calculates the mean value $\mu = (x' + y') / 2$. Next step calculates the difference value d''

$= x' - y'$. Third step extract the secret data bit by $b = \text{LSB}(d'')$. The difference value d can be restored by the Equation (2). Finally, two pixels can be restored to their original values as $x = \mu + d / 2 + \text{LSB}(d)$ and $y = \mu - d / 2$.

$$d = \begin{cases} d' + 2^{\lceil \log_2 d' \rceil + 1}, & \text{if } LM = 1 \\ d' + 2^{\lceil \log_2 d' \rceil}, & \text{if } LM = 0 \end{cases} \quad (2)$$

Arham *et al.*⁽⁷⁾ proposed data hiding scheme based on difference expansion of quad by combining the data hiding methods proposed by Alattar *et al.*⁽²⁾ and Hu et al.⁽⁴⁾. This method can not only increase the maximum embedding capacity of a quad-tuple of pixels, but also reduce image distortion caused by differential expansion and achieve recoverability by referring to certain pixel values in a quadrilateral.

In each layer hiding process, the method of Arham *et al.*⁽⁷⁾ divides an image into non-overlapping block of four pixels and let the pixels in the upper left corner be used as the reference point for the first layer of data hiding, and then the reference points for the data hiding in the second layer, the third layer and the fourth layer will be the pixels at upper right corner, the lower left corner, and the lower right corner. The purpose of turning unused pixels as reference points when each layer of data is hidden is to avoid that the amount of pixel changes caused by pixel expansion can be dispersed to each pixel, so that image distortion can be prevented from being concentrated on some pixels. Firstly, the method of Arham et al.⁽⁷⁾ separates the original image I into non-overlapping 2×2 quads, and use the left upper pixel p_1 of a quad as the reference point and represent the quad in a one-dimensional vector $P = (p_1, p_2, p_3, p_4)$. Secondly, it calculates the differences between pixels (p_{i+1}, p_i) of which $i = 1, 2, \text{ and } 3$, and produces the vector of differences $D = (d_1, d_2, d_3)$. Then the IRDE method is adopted to reduce the difference values between d_i and $d_i + 1$ in order to avoid image distortion and preserve image quality.

3. The Proposed Scheme

We present an improved method of the method proposed by Arham *et al.*⁽⁷⁾, wherein the block size was changed to $m \times n$ so that the embedding capacity of a block was increased from 3 bits to $(m \times n - 1)$ bits. Let $\mathbf{P} = (p_1, p_2, p_3, \dots, p_{m \times n})$ be the original image block. We design the scanning strategy as the search order shown in the Fig. 1. According the scanning strategy, the ordered vector of block pixels is represented as

$\mathbf{P}_\sigma = (p_{\sigma(1)}, p_{\sigma(2)}, p_{\sigma(3)}, \dots, p_{\sigma(m \times n)})$. The scanning strategy adopts image correlation for arranging the pixel vector such that the reordered adjacent pixels are associated with higher similarities.

Afterwards, the IRDE is applied to distortion and improve image quality. The scanning strategy with multi-layer recursive embedding makes each layer use a different position as a reference point, so that it can effectively control image distortion after the image is hidden behind multiple layers. Furthermore, the pixel value overflow/underflow problem can also be effectively controlled.

Here, for the sake of understanding, we give an example to illustrate its basic procedure of difference expansion. The details of data embedding/extraction phases are as follows. Assume the original image block with size of 2×3 as shown in Fig. 1.

92	96	98
95	110	120

(a)

92	96	99
95	107	127

(b)

Fig.1. A cover image block/ A stego image blocks before/after data embedding.

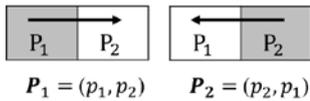
According to the scanning strategy as the search order shown in the left-upper corner of Fig. 2(c), the ordered vector of block pixels is represented as $\mathbf{P}_\sigma = (p_{\sigma(1)}, p_{\sigma(2)}, p_{\sigma(3)}, \dots, p_{\sigma(m \times n)})$. That is the block of ordered pixels with scanning strategy is $\mathbf{P}_\sigma = (p_{\sigma(1)}, p_{\sigma(2)}, p_{\sigma(3)}, p_{\sigma(4)}, p_{\sigma(5)}, p_{\sigma(6)}) = (92, 96, 98, 120, 110, 95)$. Calculate the differences between each adjacent pixels $p_{\sigma(i)}$ and $p_{\sigma(i+1)}$ and obtain the difference vector $\mathbf{D} = (d_{\sigma(1)}, d_{\sigma(2)}, d_{\sigma(3)}, \dots, d_{\sigma(m \times n - 1)})$. In this case, $\mathbf{D} = (d_{\sigma(1)}, d_{\sigma(2)}, d_{\sigma(3)}, d_{\sigma(4)}, d_{\sigma(5)}) = (4, 2, 22, 10, 15)$. Next calculate $n_i = \lceil \log_2 d_{\sigma(i)} \rceil$ that present how many number bits of secret data can be hidden in the difference values. So these n_i 's are (2, 1, 4, 3, 3). Apply Equation (1) to calculate the reduced differences as $\mathbf{D}' = (2, 1, 14, 6, 7)$. And record the extra

information for image restoration. $\mathbf{LM} = (0, 0, 0, 0, 0, 1)$. Assume the to-be-hidden message is (0, 1, 1, 1, 1). The embedding steps of IRDE include expanding the new difference values, embedding one secret bit as LSB substitution and retuning the ordered pixel vector by $q_1 = q_{\sigma(1)} = p_{\sigma(1)}$ and $q_{\sigma(i+1)} = p_{\sigma(i)} + d_{\sigma(i)}$. Where $q_1 = q_{\sigma(1)} = p_{\sigma(1)}$ is the reference point that remains unchanged before/after data embedding. $\mathbf{Q}_\sigma = (92, 92 + 4 = 96, 96 + 3 = 99, 98 + 29 = 127, 120 - 13 = 107, 110 - 15 = 95)$. Accordingly, $\mathbf{D}'' = (4, 3, 29, -13, -15)$ and $\mathbf{Q} = (92, 92 + 4 = 96, 96 + 3 = 99, 98 + 29 = 127, 120 - 13 = 107, 110 - 15 = 95)$ as Fig. 1(b).

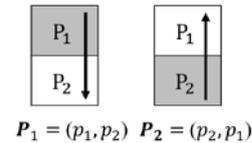
The process of the data extraction and image recovery is illustrated as follows.

First, the receiver should have the extra information \mathbf{LM} , and divide the image into $m \times n$ non-overlapping blocks. Guided by the scanning strategy, the receiver can get an ordered stego-pixel vector for each block. Calculate the difference values, extract the secret data by LSB substitution and recovery the expanded differences using \mathbf{LM} and restore the original pixels. As the example mentioned above, $\mathbf{LM} = (0, 0, 0, 0, 0, 1)$, $\mathbf{Q}_\sigma = (92, 96, 99, 127, 107, 95)$ and the scanning strategy is in 2×3 search order with the p_1 as the reference point. According the steps illustrated below, the data extraction and image recovery can be completed.

- (1) $p_1 = q_1 = 92$
- (2) $d_1'' = |96 - 92| = 4$; $b_1 = \text{LSB}(4) = 0$; $d_1' = \text{floor}(4 / 2) = 2$; $d = 2 \times d_1' = 4$; $p_2 = 92 + 4 = 96$.
- (3) $d_2'' = |99 - 96| = 3$; $b_2 = \text{LSB}(3) = 1$; $d_2' = \text{floor}(3 / 2) = 1$; $d = 2 \times d_2' = 2$; $p_3 = 96 + 2 = 98$.
- (4) $d_3'' = |127 - 98| = 29$; $b_3 = \text{LSB}(29) = 1$; $d_3' = \text{floor}(29 / 2) = 14$; $d = 22$; $p_4 = 98 + 22 = 120$.
- (5) $d_4'' = |107 - 120| = 13$; $b_3 = \text{LSB}(13) = 1$; $d_4' = \text{floor}(13 / 2) = 6$; $d = 10$; $p_5 = 120 - 10 = 110$.
- (6) $d_5'' = |95 - 110| = 15$; $b_3 = \text{LSB}(15) = 1$; $d_5' = \text{floor}(15 / 2) = 7$; $d = 15$; $p_6 = 110 - 15 = 95$.



(a) block size: 1×2



(b) block size: 2×1

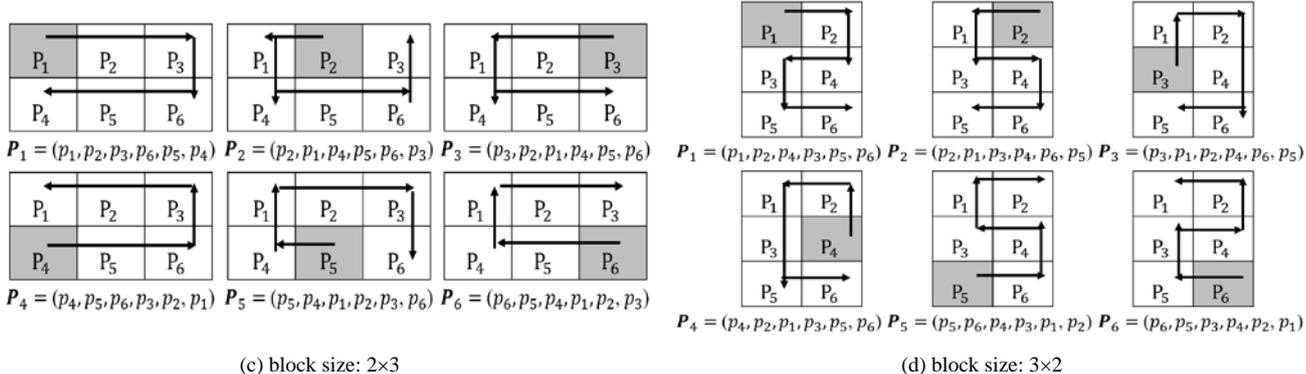


Fig. 2. Scanning Strategy for various block sizes and the pixel marked as gray is a reference point.

4. Experimental Results

This experiment uses six standard 512x512 images including Airplane, Baboon, Barbara, Boat, Lena, and Peppers. When the threshold (T) is set as 30, the proposed method with different block sizes 1x2, 2x1, 2x3, 3x2, 3x3, 4x4, and 5x5 was conducted for comparison with the performance proposed by Arham *et al.* Each block will have a reference point, and the reference point will not be hidden with data. When the block is divided larger, the smaller the number of blocks representing the entire image, the less reference points cannot be hidden, so the more pixel values can be hidden. Conversely, when the block is smaller, the more the number of reference points of the entire image, so the number of embeddable areas is lower. The results are shown in Table 1 where when the block size is 1x2, the average payload is 129,947 bis and average embedding capacity is 0.496 bit per pixel (bpp). Fig. 3 shows the PSNR values of various images compared to those of Arham *et al.*'s method for fixed embedding capacity. We can see when the block size is 2x1, the image quality of Baboon with line chart

in bright pink of Fig. 3(b) is down to 37 dB and is worse than that of other methods, which indicates that the quality is unsatisfactory for the complex images with the complicated textures by applying the scanning strategy of vertical order. As a whole, the performance is better when the block is larger.

Finally use multiple layers of possession to increase embedding capacity. In order to avoid having the value expanding on the same pixel each time when the embedding procedure is applied, leading to overflow/underflow issues, each block uses different block pixels as reference points.

This paper conducted 9-layer embedding to compare the performance of different block sizes. The comparison of the embedding capacity (bpp) in multi-layer embedding ($T = 30$) is shown in Fig. 4. We can see that when the block is 2x3, the overall performance is the best. The cover images and stego images using 9-layer embedding method are shown in Fig. 5. When the block is smaller, the stego image has higher image quality, and when the block is bigger, it has higher embedding capacity. Therefore, these results of Fig. 4 can offer the selection with appropriate block size according to the demand in practical applications.

TABLE 1. Comparison of the Actual Performance in Terms of Payload and Embedding Capacity ($T = 30$).

	Payload (bits)								Embedding capacity (bits per pixel)							
	Arham et al.	Method with size of 1x2	Method with size of 2x1	Method with size of 2x3	Method with size of 3x2	Method with size of 3x3	Method with size of 4x4	Method with size of 5x5	Arham et al.	Method with size of 1x2	Method with size of 2x1	Method with size of 2x3	Method with size of 3x2	Method with size of 3x3	Method with size of 4x4	Method with size of 5x5
Lena	196,083	130,763	130,953	217,150	217,201	230,734	245,232	249,139	0.748	0.499	0.500	0.828	0.829	0.880	0.935	0.950
Baboon	193,314	130,427	125,122	214,754	212,967	227,689	242,550	246,742	0.737	0.498	0.477	0.819	0.812	0.869	0.925	0.941
Barbara	191,466	127,877	130,173	213,054	213,896	226,659	240,624	244,430	0.730	0.488	0.497	0.813	0.816	0.865	0.918	0.932
Boat	194,583	129,614	130,884	215,529	216,014	229,123	243,525	247,179	0.742	0.494	0.499	0.822	0.824	0.874	0.929	0.943
Airplane	195,282	130,246	129,996	216,291	216,376	229,988	244,445	248,461	0.745	0.497	0.496	0.825	0.825	0.877	0.932	0.948
Peppers	195,414	130,756	130,662	217,017	217,003	230,567	245,124	249,017	0.745	0.499	0.498	0.828	0.828	0.880	0.935	0.950
Average	194,357	129,947	129,632	215,633	215,576	229,127	243,583	247,495	0.741	0.496	0.495	0.823	0.822	0.874	0.929	0.944

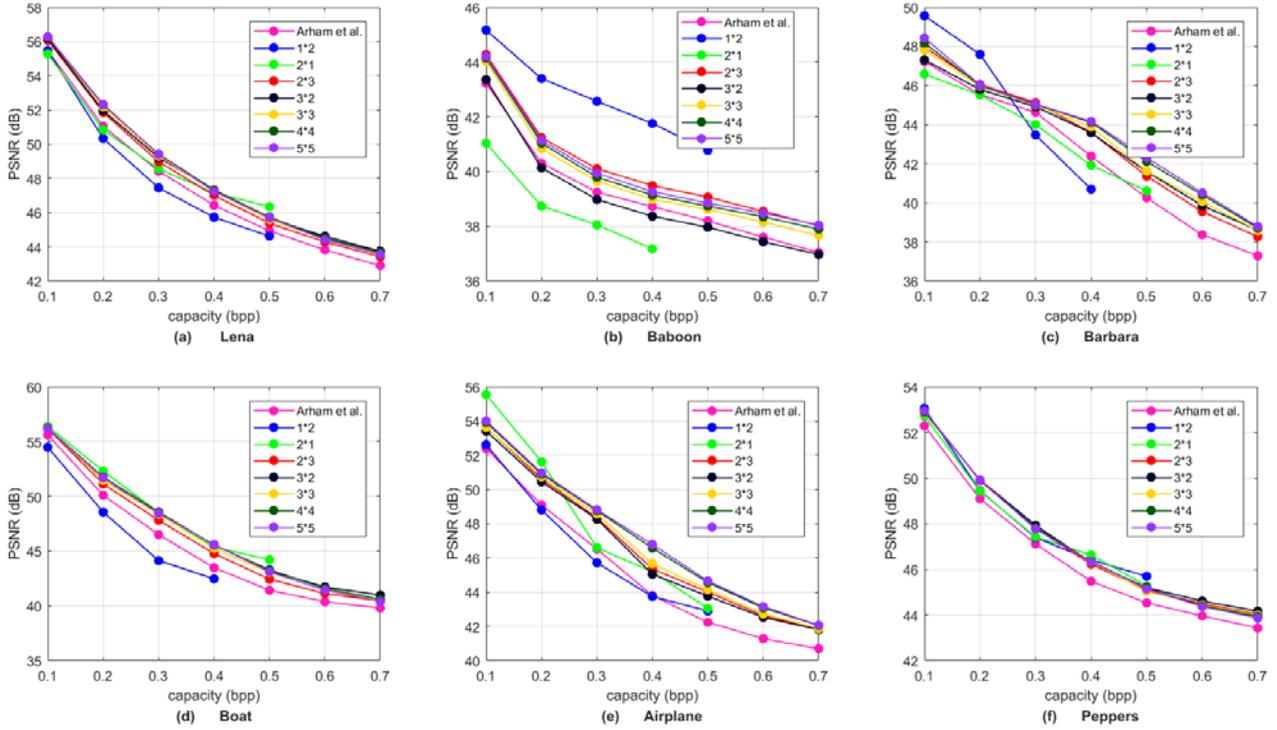


Fig. 3. Compare the image quality of a single layer for a given embedding capacity ($T = 30$).

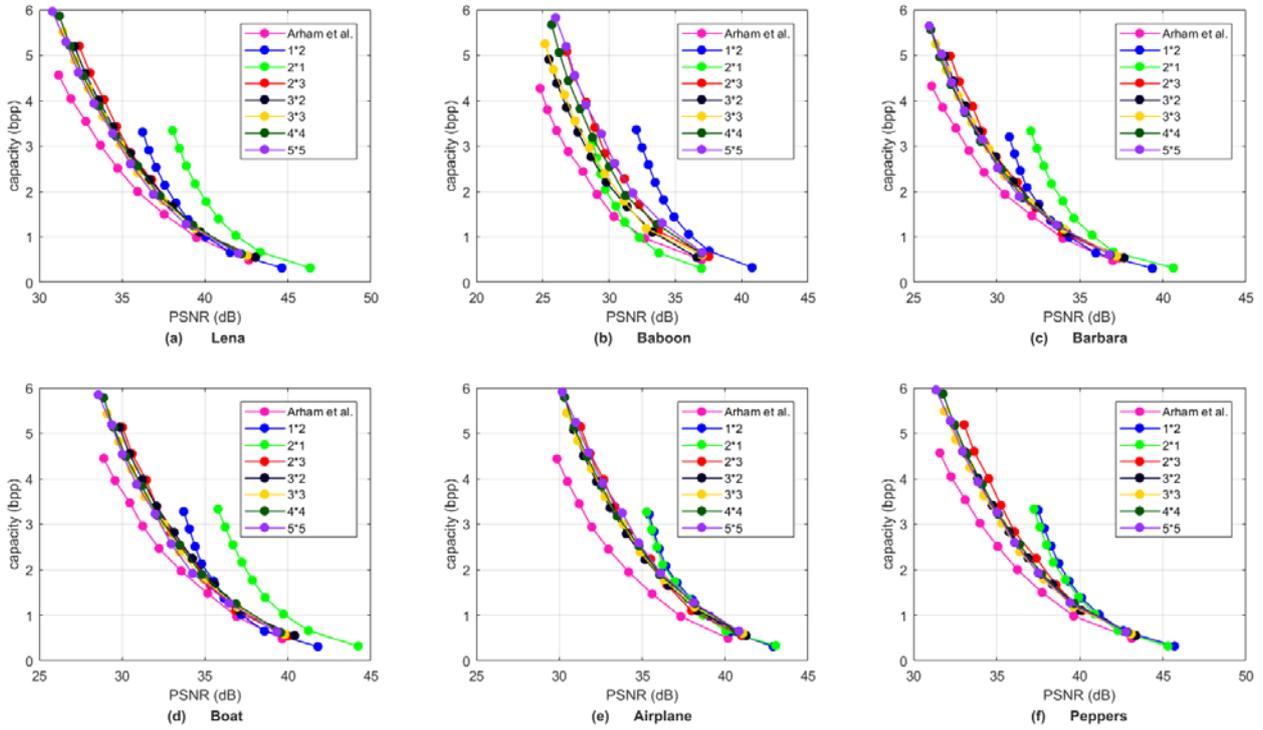


Fig. 4. Comparison of the embedding capacity (bpp) in multi-layer embedding ($T = 30$).

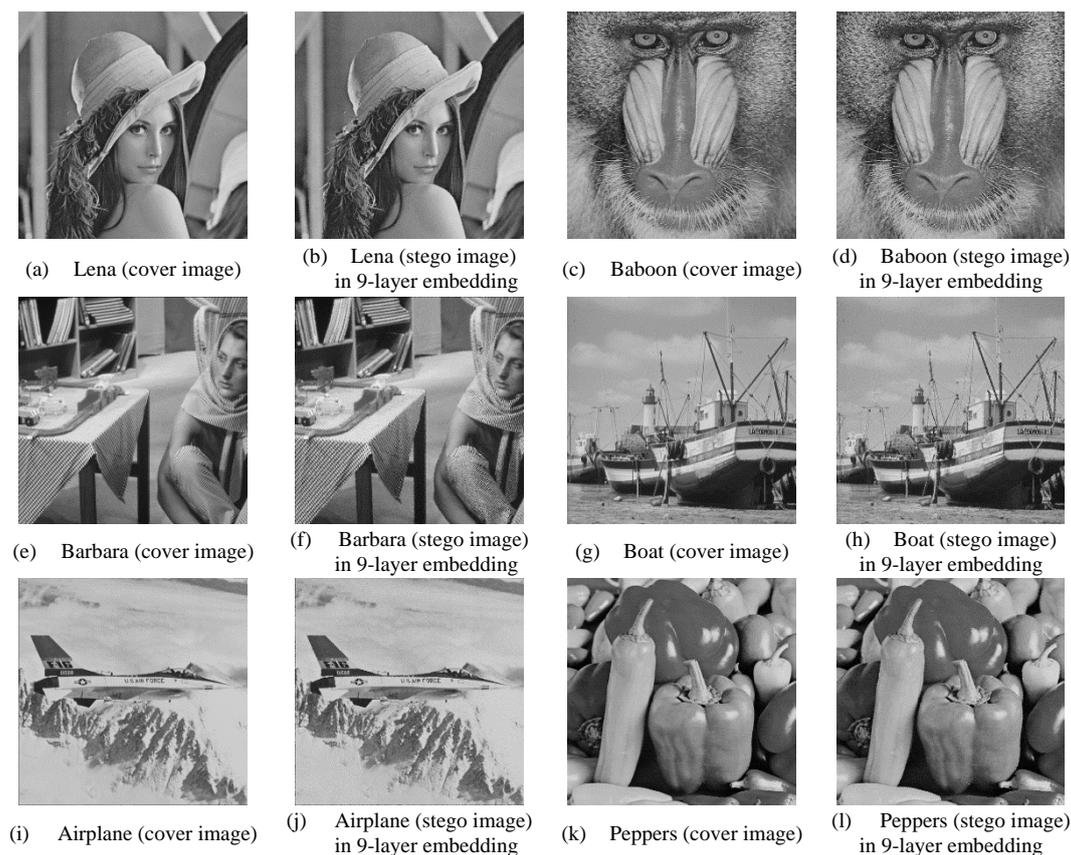


Fig. 5. Six cover images and stego images (block size: 2×3)

5. Conclusions

This paper presents an improved method of the method proposed by Arham *et al.*⁽⁷⁾, wherein the block size was changed to $m \times n$ so that the embedding capacity of a block was increased. Based on the correlation of neighboring pixels, different scanning strategies are used for multi-layer recursive message embedding in the message hiding process of blocks of different sizes. Then IRDE method is conducted to reduce the difference expansion and avoid distortion of overflow/underflow. The experimental results indicate that the smaller the block, the lower the hidden amount, the larger the block, and the higher the hidden amount. This paper can use the appropriate block size for hiding according to the purpose of use, so as to achieve the best performance.

Acknowledgment

This research was partially supported by the Ministry of Science and Technology of the Republic of China under the Grants MOST 106-2221-E-324-006 -MY2.

References

- (1) Jun Tian, "Reversible Data Embedding Using a Difference Expansion," *IEEE Transactions on Circuits Systems Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.
- (2) Adnan 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.
- (3) Chiang-Lung Liu, Der-Chyuan Lou, Chien-Chung Lee, "Reversible Data Embedding Using Reduced Difference Expansion," *Proceedings of the third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP 2007)*, pp. 433–436, 2007.
- (4) Yi Hu, Wei Song, and Jianjun Hou, "Improved Reduced Difference Expansion Based Reversible Data Hiding Scheme for Digital Images," *Proceedings of the 9th International Conference on Electronic Measurement & Instruments (ICEMI '09)*, pp. 4–315, 2009.

- (5) X. Wang, X. Li, B. Yang, and Z. M. Guo, "Efficient Generalized Integer Transform for Reversible Watermarking," *IEEE Signal Processing Letters*, vol. 17, no. 6, pp. 567-570, June 2010.
- (6) Y. Qiu, Z. Qian, and L. Yu "Adaptive Reversible Data Hiding by Extending the Generalized Integer Transformation," *IEEE Signal Processing Letters*, vol. 23, no.1, pp. 409- 413, Jan. 2016.
- (7) Aulia Arham, HanungAdi Nugroho, and TeguhBharata Adji, "Multiple Layer Data Hiding Scheme based on Difference Expansion of Quad," *Signal Processing*, vol. 137, no. 8, pp. 52-62, 2017.