NUEJ
Naresuan University
Engineering Journal

A Genetic Algorithm-Based Reversible Data Hiding Approach for Enhancing QR Code Security

Chaiyaporn Panyindee*

Department of Computer Engineering, Faculty of Engineering, Rajamangala University of Technology Rattanakosin, Nakhon Pathom, 73120, Thailand

*Corresponding author e-mail: Chaiyaporn.pan@rmutr.ac.th

(Received: 15 March 2003, Revised: 6 June 2023, Accepted: 30 October 2023)

Abstract

The problems concerning efficient RDH algorithms are often complex and involve a combination of several methods. Embedding capacity and each image require different optimum parameters. This paper presents an investigation of the parameters for the reversible data hiding algorithm for QR code images. One tool was used for finding those optimal parameters. The genetic algorithm was applied to find the weighting value and level of Expanded Variance Mean sorting that provides the lowest possible distortion for each image and each embedding capacity. Using pixel sorting before embedding is essential for modern RDH algorithms to reduce the location map size, thus allowing more information to be embedded with less distortion. The path of the threshold values of the QR code image (close to 0 and 255) was also checked to ensure the best embedding interval. The performance test for the proposed method used six QR code images (Image with the smallest and least variance to image with the largest and most variance), and the hidden bits were random. The experimental results of the proposed method show that the peak signal-to-noise ratio values are superior compared to the previous two works, averaging about 2 dB compared to LV+EMSW and 0.5 dB compared to EVM+EMSW. In the future, it should be possible to explore multi-bit embedding schemes for smooth areas that enable more embedding and still have low distortion.

Keywords: genetic algorithm, Linear multi-scale weighting, reversible data hiding.

1. INTRODUCTION

Currently, QR codes are used extensively in media applications. The information transmitted with the image QR code can be easily accessed by scanning by using a smartphone. However, some information still requires privacy, such as transaction numbers, date of birth, national ID number, etc. For this reason, reversible data hiding (RDH) techniques are considered preventing unauthorized access to top-secret information.

RDH is part of steganography, which has become popular among image processing researchers in recent years. The key condition of RDH is to restore both the original host image and message without loss. In addition, RDH methods should have a high embedding capacity, with less distortion. The main challenge faced by RDH in QR code imaging is trying to hide information in a pixel with high variance and intensity values close to the maximum and minimum without causing the Overflow and Underflow problem. The first RDH was presented by Mintzer et al. (1997). They introduced a visible embedding technique exploiting the property of reversibility of the original image. Fridrich et al. (2002) proposed a lossless compression algorithm for RDH. A least significant bit (LSB) substitution technique using an efficient entropy coder is proposed by Celik et al. (2002). Yang et al. (2004) proposed an integer discrete cosine transform (DCT) based RDH for images using a companding technique. Several RDH techniques based on integer discrete wavelet transform (DWT) are proposed by Xuan et al. (2002-2004). Zou et al. (2006) proposed a semi-fragile, lossless digital watermarking scheme based on integer DWT.

Tian (2003) introduced reversible data embedding using a difference expansion (DE). His scheme divides the image into pairs of pixels and uses each pair to hide a bit of information. Thus, his embedding capacity is at best 0.5 bits/pixel. The major problem in the original DE method was the large size of the location map, which must be included as part of the payload. Therefore, reducing the size of the location map is one of the key goals in this field. Kamstra and Heijmans (2005) improved the DE method by sorting pairs according to correlation measures facilitating compression. Kim et al. (2008) reduced the size of the location map by removing non-ambiguous parts. Their methods achieve better results when compared to Kamstra and Heijmans' method (2005). Ni et al. (2006) introduced a histogram shifting technique in the spatial domain. Thodi and Rodrıguez (2007) explored the histogram shifting (HS) method by employing prediction errors for efficiency, which is sometimes called prediction-error histogram shifting (PEHS). Sachnev et al. (2009) improved the performance of PEHS by using sorting. Kotvicha et al. (2012) improved the performance of the sorting using expand variance mean (EVM) technique base on Sachnev's algorithm etc. For the past few years, a new framework based on PEHS called pixel-value-ordering (PVO) has drawn significant interest from researchers. PVO was initially proposed by Li et al. (2013) to embed secret data into the minimum and maximum pixel values of divided image blocks. After that, Ou et al. (2014) attempted to exploit the correlation of smooth blocks, so they proposed PVO-k to modify the blocks that have the k largest or smallest pixels. In 2014, Peng et al. (2014) embedded secret data into smooth blocks by utilizing the spatial position and proposed an improved PVO (IPVO). In 2015, Wang et al. (2015) proposed a PVO method in which block sizes are dynamic. In the same year, Qu and Kim (2015) proposed a pixel-based PVO method to predict pixels one by one. In 2016, the pairwise modification was employed in PVO, and unique 2D mapping was designed according to the specific 2D PEHS by Ou et al. (2016). In subsequent years, pairwise PVO was improved by Dragoi et al. (2018), Gao et al. (2019), He and Cai (2020), and Zhang et al. (2020) to achieve better performance. In 2021, Fan et al. (2021) merged the advantages of sorting and averaging by combining IPVO and a conventional Rhombus predictor through the multi-predictor mechanism. Their method outperformed state-of-the-art PVO-based RDH methods. In the same year, Panyindee (2021) improved sorting and conventional Rhombus predictor by modifying the dynamic variance mean (DVM) sorting models and applying a fitting weight rhombus (FWR) predictor. Recently, Fan et al. (2023) proposed one-dimensional and two-dimensional flexible patch moving (FPM) modes to move the patch with adaptive step lengths to further improve the embedding performance.

Over the past few years, RDH methods (2015-2019) have been applied to QR code images, causing Overflow and Underflow problems because the pixels in QR code images are the highest and lowest values (255 and 0). To solve this problem, Panyindee et al. (2019) proposed the linear multi-scale weighting (LMSW) technique; their approach achieves several embedding capacities. Another problem with QR code images is that the pixel texture is highly variable. As a result, the use of local variance sorting is insufficient for correct pixel sorting (Low PE should be before high PE). In the same year, Panyindee et al. (2019) applied the EVM technique together with the LMSW technique based on PEHS, their method achieved PSNR results superior to previous methods, especially with small payload embedding.

However, when a RDH algorithm (2019) combines several methods together, there are several parameters that must be repeated (find) in many parts, such as weighting value (W), threshold values (T_n, T_p), EVM value (S) in order to get the best possible PSNR result. Their parameter search uses step-by-step tuning for each embedding capacity, which is difficult because many parameters must be searched simultaneously as unknown. In this paper, GA was used to find the optimal parameters for each image and each embedding capacity. Our results provide low distortion and achieve PSNR values that are superior to the two previous works.

The rest of the paper is arranged as follows. Section 2 discuss the analyze the modifying pixels using PEHS + sorting for QR code images problem. Section 3 describes the proposed method. Section 4 shows the experimental results and concluding remarks are given in the final section.

2. Analyze The Overflow Problem Of QR Codes

Several predictors based on PEHS (2007) were proposed to reduce distortion significantly, such as median edge detector predictor (MED) (2007) and (2009), the gradient-adjusted predictor (GAP) (2011), the rhombus predictor (2019), accurate predictor (2012), PDE predictor (2013), Gaussian weight predictor (2016), etcetera. PEHS encoding can be calculated as follows.

$$D_{i,j} = \begin{cases} 2d_{i,j} + b, & \text{if } d_{i,j} \in \left[T_n; T_p\right] \\ d_{i,j} + T_p + 1, & \text{if } d_{i,j} > T_p \text{ and } T_p \geq 0 \\ d_{i,j} + T_n, & \text{if } d_{i,j} < T_n \text{ and } T_n < 0 \end{cases} \tag{1}$$

Where d is the PE value. b is the embedded data bit (i.e., 0 or 1). T_n is the negative threshold value. T_p is the positive threshold value. The determination of the T_n and T_p range directly affects the embedding capacity, and the obtained PSNR values. Therefore, the range of T_n and T_p should fit the required payload size. Conversely, if Tn and T_p are extended more than necessary, the PSNR is dropped. PEHS decoding was calculated by equations (2) and (3).

$$d_{i,j} = \begin{cases} \left\lfloor D_{i,j} / 2 \right\rfloor, & \text{if } D_{i,j} \in \left[2T_n; 2T_p + 1\right] \\ D_{i,j} - T_p - 1, & \text{if } D_{i,j} > 2T_p + 1 \text{ and } T_p \geq 0 \\ D_{i,j} - T_n, & \text{if } D_{i,j} < 2T_n \text{ and } T_n < 0 \end{cases} \tag{2}$$

$$b = D_{i,j} \bmod 2, \quad D_{i,j} \in [2T_n; 2T_p + 1] \tag{3}$$

Analyzes of a recent problem of RDH base on PEHS for QR code images, where the gray-scale values were the highest and lowest as shown in Figure 1.

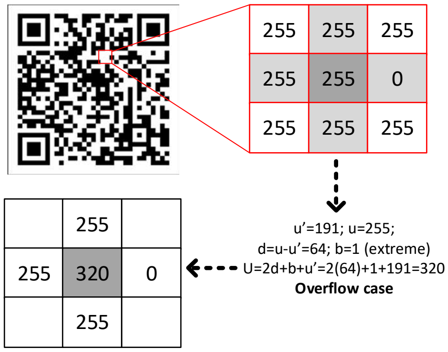


Figure 1 An example of the overflow problem after data embedding using PEHS.

These pixels often caused Overflow and Underflow problems after the modified pixels using condition (4).

$$0 > U = 2d + b + u' > 255 \qquad (4)$$

Panyindee et al. (2019) introduced a LMSW technique to create free space for each size of embedding data, but still able to scan and access the data. The weighting (W) value can be calculated as Equation (5).

$$W = \frac{255 - C}{P_{MAX}} \quad ; \quad C \in \{1, 2, 3, ..., 255\} \qquad (5)$$

Where C is the ascending constant, P_{MAX} is the maximum gray-level intensity, I_m is an original QR code image and the weighted image I_w can be calculated as

$$I_W = \lceil W \times I_m \rceil \qquad (6)$$

Their approach makes it possible to hide information in QR code images by a PEHS method. However, finding the ideal W parameter of the RDH algorithm for each image and each embedding size is still necessary and must be investigated.

Another enhancement of the sorting process of RDH algorithm (2019), is that the static sorting technique (local variance (2009)) that has been replaced by an adaptive sorting technique (EVM (2012)). The EVM is applied to get the correct sequence of PE values in each QR code image and each embedding size. EVM can be calculated as follows.

$$\bar{\mu}_{i,j}^{S} = \frac{\sum_{l=-\lfloor S/2 \rfloor}^{\lfloor S/2 \rfloor} \mu_{i+l,j+l} + \sum_{k=1}^{\lfloor S/2 \rfloor} \sum_{l=-\lceil S/2 \rceil}^{\lceil S/2 \rceil - 2k} (\mu_{i+l,j+l+2k} + \mu_{i+l+2k,j+l})}{2\lceil S/2 \rceil^{2} + 2\lceil S/2 \rceil + 1 - 4(S \bmod 2)} \qquad (7)$$

Where μ is the local variance. S is the degree of expanding μ. l is the length radius of the mask. Note, that the correct PE sequencing significantly increases the PSNR value. When a low PE is used, the post-embedded distortion remains as low as the PE itself. The QR code image looks like noise. Experiments have shown that the dynamic data sorting techniques produce excellent results, especially with highly variable images.

Two previous RDH solutions for QR code images make data hiding possible. However, the consequence is that their work has more parameters that must be searched for optimal results. Thus, a tool to find the optimal parameters became necessary to get as close to the best possible results. In this work, GA was considered.

3. THE PROPOSED METHOD

From solving the RDH problem for the QR code image to the improvement of the embedding method, an algorithm (2019) that combines techniques such as PEHS, EVM, Two-pass tasting (TPT), Double Embedding for QR code images was investigated to get the parameters that achieve the highest possible PSNR value. In this work, GA is applied to find those suitable parameters. GA is an ideal tool for finding optimal solutions using natural selection or natural evolution. Moreover, default parameters are also set using the best parameters of the former methods to ensure that the results are not worse than the previous methods. An overview of the application of GA in combination with the RDH algorithm is shown in Figure 4. To ensure the efficacy of the proposed method, six QR code images representing the smallest image with the least variance to the largest image with the greatest variance were used for testing. The data bits used for embedding were random, starting with 10,000 bits then increasing to 20,000 bits and 30,000 bits, respectively, until embedding was not possible. The measurement tools use PSNR and MSE values, which are standard formats for RDH work.

GA is a method for solving both unconstrained and constrained optimization problems based on natural selection, the procedure that runs biological evolution. GA repeatedly modifies a population (or chromosomes) of individual solutions. In each step, GA selects individuals from the current population to serve as parents and uses them to produce children for the subsequent generation. Over successive generations, the population evolves into the best solution. GA has parameters known as "gene" and "chromosome". Long genes produce efficient information exchange results. Our chromosome was a row of binary numbers which consists of two genes, as shown in Figure 2.

$b_{m_W-1}b_{m_W-2}\ldots b_{0_W}$	$b_{m_S-1}b_{m_S-2}\ldots b_{0_S}$
gene1=W	gene2=S

Figure 2 A chromosome in GA consists of two genes.

Binary numbers can be changed to decimal numbers by using equation (8)

$$decimal = l_b + (u_b - l_b)(\sum_{i=1}^{m} 2^{i-1} b_{i-1}) / (2^m - 1) \quad (8)$$

Where u_b is the upper bound parameter. l_b is the lower bound parameter. m is the number of bits used for each gene. b_{i-1} is a binary value of i^{th}, after creating n rows of different chromosomes. An objective function (Sometimes called the fitness function) is defined as the mean square error (MSE) calculated as equation (9).

$$MSE = \sum_{i=1}^{N} \sum_{j=1}^{M} \left| u_{i,j} - U_{i,j} \right|^2 / (N \cdot M) \quad (9)$$

MSE is the total error between the original image and the embedded image. GA will search for W, and S which has the minimum MSE in each image and each required payload. To increase the performance of GA, the scope of the parameters is defined: by setting the default weight W_{Init} = 0.992, while the maximum weighting is W_{max} = 0.921, which is superior for large data embedding. For T_n and T_p are determined at -1 and 0, respectively, according to the required payload size. Note, that the histogram of prediction error of QR code images is not Laplacian, as shown in Figure 3. Most PE values in the first layer (cross layer) were 0 whereas in the next layer (dot layer) the PE value graph dropped rapidly as the pixels in the cross layer were already embedded and used to make predictions in the dot layer. This embedding process, called Double Embedding (2009), which can be effectively used with the rhombus predictor and sorting techniques. The combination of many techniques allows the embedding of a large payload to be possible. Another note, for most PE values in the dot layers found are 0 and -1 drop down, which directly affected the high-distortion after the embedded data had been extracted. It became the reason that threshold values were fixed at those two values. The last parameter is S, from the test, S_{max} = 64 is enough for the extreme version of the QR code image (i.e., version 40). All the reasons mentioned above will help GA run faster.

The next step will be chromosomal enhancement using three steps of a genetic operation: Selection, Crossover, and Mutation as shown in Figure 4, which also shows an overview of the system. After all operations are completed, the next generation of chromosomes is regenerated. New chromosomes often provide better or equal fitness values. Although the algorithm cannot

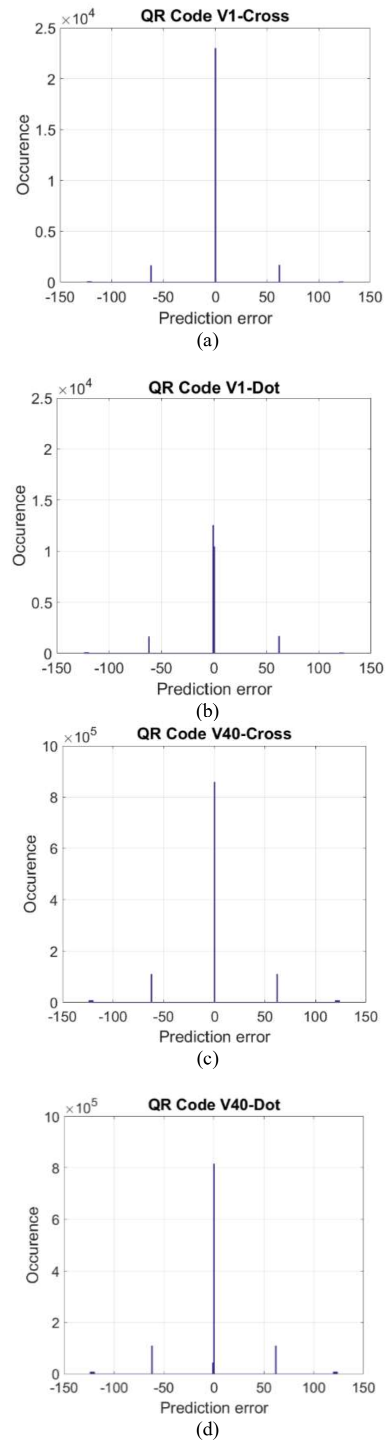


Figure 3 (a), (c) shows histogram of rhombus predictor PEs in cross-layer and (b), (d) shows histogram of rhombus predictor PEs in dot-layer of the three QR Code images.

guarantee that the best results will be found, it will approach the best potential value. Sometimes the number of generations is another important factor to get the best results.

To reduce embedding distortion, all pixels are examined status by the TPT definition (2016), which are separated into seven sets: *EE*, *ES*, *SS*, *E*, *S*, *NE*, and *NS*. Where $EB(T_n,T_p)$ is an embeddable set while $LM(T_n,T_p)$ is a non-embeddable set and must be mapped and are defined as:

$$EB(T_n,T_p) = \{(i,j) \in EE \cup ES,\} \quad (10)$$

$$LM(T_n,T_p) = \{(i,j) \in E \cup S \cup NE \cup NS\} \quad (11)$$

Therefore, the number of pixels that can be embedded is equal to the number of *LM* plus the number of the payload, as in Equation (12).

$$n\left(EB(T_n,T_p)\right) = n\left(LM(T_n,T_p)\right) + n\left(Payload\right) \quad (12)$$

Detailed encoding and decoding algorithms are described as follows.

3.1 Encoding Algorithm

1. Start for GA by the initial population setting at *n* chromosomes (*W* and *S*).

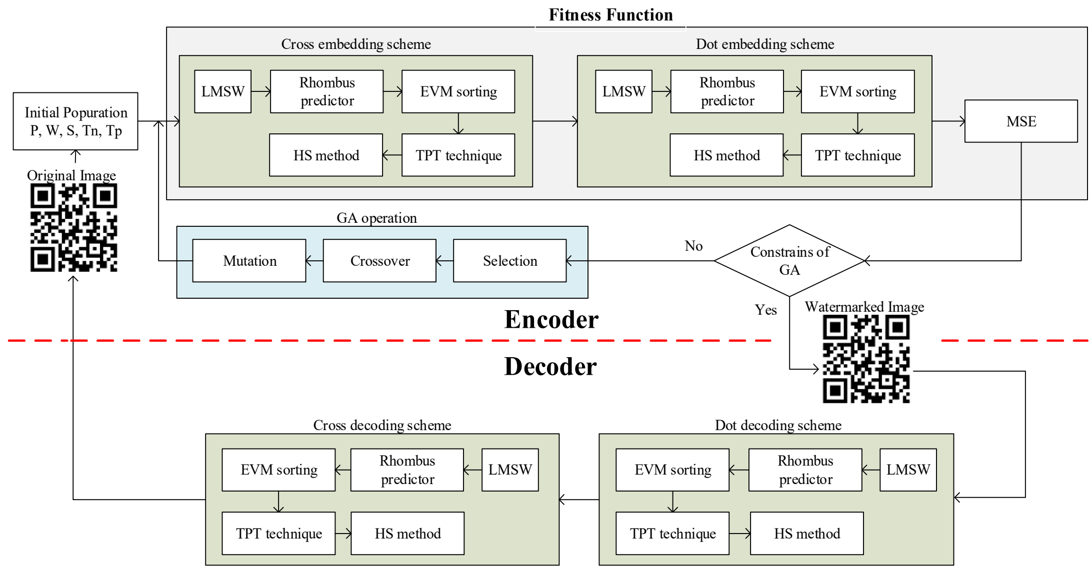


Figure 4 Block diagram for encoding and decoding algorithms using GA.

2. Convert all values to binary from decimal numeric parameters using equation (8).
3. Determine the cross and dot layer position in the QR code image according to Sachnev et al. (2009).
4. Each layer is calculated as follows.
   4.1 Starting from the position in the cross layer, calculated as follows:
      4.1.1 the prediction value $u' = floor[(v_{i,j-1} + v_{i,j+1} + v_{i-1,j} + v_{i+1,j})/4]$ and $d = u - u'$.
      4.1.2 Sort all pixels according to the EVM technique using Equation (7).
   4.2 Check the embedding area (Count the number of sets *EB* and *LM*) using Equation (12) until the desired position is obtained, but if not found, go to step 1 again.
5. For each chromosome that was previously collected, let start embedding and calculate the results following steps 5.1-5.5.
   5.1 Keep the LSB of the first h pixel and replace it with a header file for data recovery.
   5.2 Store the location map using Equation (11) and insert the LSB from the previous step into a request payload.
   5.3 Embed data using equation (1) and modified pixel value $U_{i,j} = u'_{i,j} + D_{i,j}$.
   5.4 Apply the above steps in a dot layer.
   5.5 Calculate the *MSE* value using Equation (9).
6. If GA is run up to the last generation, the lowest MSE received in that current generation, that is the best fitness value.

For data recovery, it is necessary to have the parameters transmitted from the encoding process to the decoder, which are stored in the header file as: 10 bits for a weighted value *W*, 6 bits for an expansion level *S*, 2 bits for a negative threshold T_n, 2 bits for a positive threshold T_p, and 21 bits for the size of a request payload *P*, a total of 41 bits. The decoding process is as follows.

3.2 Decoding Algorithm

1. Separate all positions in the embedded QR code image into crosses and dot layers according to their encoding.
2. Extract the LSB of the first *h* pixel of header file in the dot layer and convert all parameters (*W*, T_n, T_p, *S* and *P*) from binary to decimal using equation (8).
3. Calculate the *EVM* of every pixel and sorts them in ascending order.

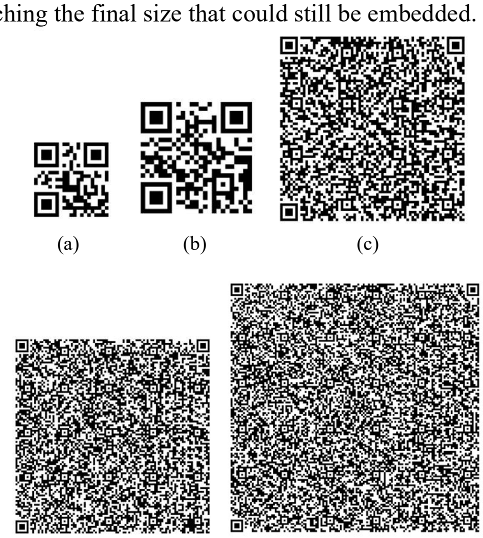
4. Calculate prediction values and PE values of all pixels using the rhombus predictor and Equation (2) respectively.
5. Check the status of each PE values and threshold values from a previously header file using definition 2 in TPT.
6. Separate pixels that are *EE* and *ES* sets sequentially to restore the original pixels using Equation (3) and replace the original LSB of the header file back to the first *h* pixel.
7. Repeat steps 2 - 6 in the cross layer.

4. Experimental Results

In the experiment, six different versions of the QR code image were used to test the effectiveness of the proposed method, which was created online by Nayuki (2021) and set the standard according to LV+LMSW (2019) as shown in Figure 5. Note that all six images were RGB color images with the same gray-scale for all three planes, so the embedding performance test used only one plane. All processes were tested by using a PC CPU Core I7-7700 HQ 2.8 GHz, 4 GB RAM, 256 GB HDD on Windows 10 Pro 64 bit operating system, processed by using MATLAB version R2016a. The PNSR value was used to measure the performance of the proposed method, which was calculated as follows.

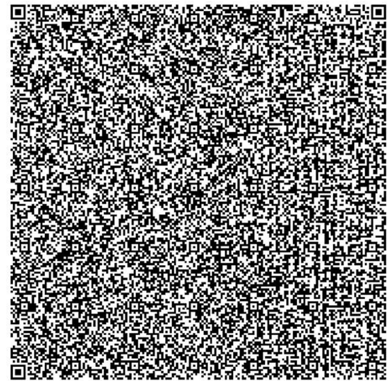
$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \quad (13)$$

Several randomized bits were used in testing by starting at 10k bits and increasing to 20k, 30k, respectively, until reaching the final size that could still be embedded. The



(a) (b) (c)

(d) (e)

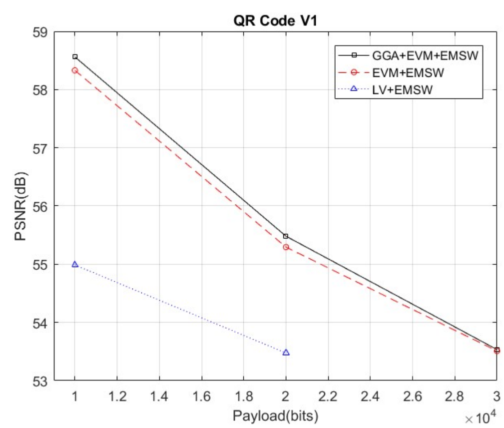


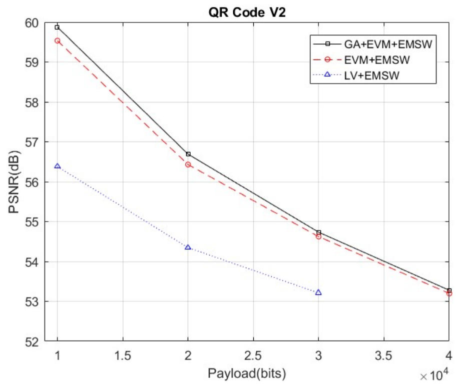
(f)

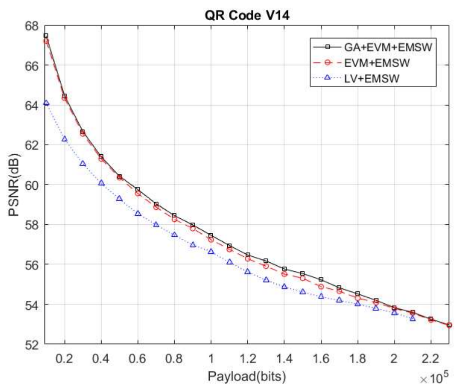
Figure 5 Test images: (a) QR Code V1 size 232 x 232 pixels, (b) QR code V2 size 264 x 264 pixels, (c) QR code V14 size 648 x 648 pixels, (d) QR code V22 image size 904 x 904 pixels, (e) QR code V31 image size 1192 x 1192 pixels, (f) QR code V40 size 1480 x 1480 pixels.

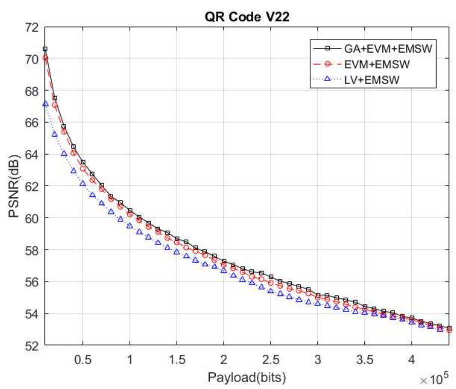
parameters *W* and *S* for each image and each embedding size investigated by GA are shown in Table 1. Note that image size and pixel variability affected the size of the embedding. In the image, the QR code V1 could only embed a maximum of approximately 30k bits, while the QR code V40 could embed data up to around 1,160k bits. However, the average bit-embedding rate of all six images remained about 0.5 bpp.

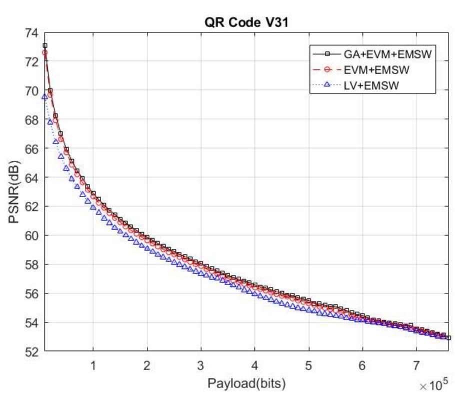
The proposed method provided superior PSNR results than two previous methods (2019) for almost every size of the payload with six QR code images, as shown in Figure 6. The combination of GA+EVM+LMLW based











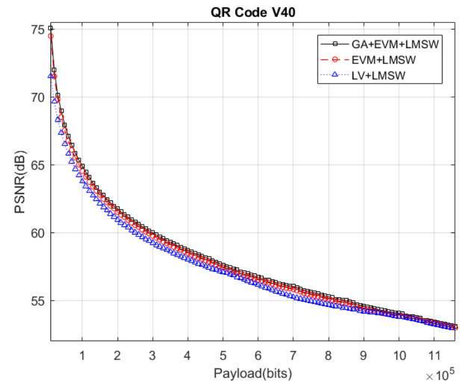


Figure 6 Graphs showing PSNR VS Payload values of the proposed method (GA+EVM+LMSW) compared with two previous methods (EVM+LMSW) and (LV+LMSW) for the six different QR code images.

on PEHS achieved the best possible results in a limited range. Note that sorting techniques are essential and must be part of the RDH algorithm, especially with highly variable images, to achieve low distortion. From the experiments, dynamic sorting techniques yielded superior results to static sorting techniques. In Figure 6, the PSNR graph shows a solid black line (GA+EVM+LMSW) and a red dotted line (EVM+LMSW) comparatively above the blue dotted line (LV+LMSW), especially in the early stages of embedding at low payloads. In the footer of embedding large payloads, the PSNR graphs of the three methods were seldom different because the PE values in the dot layer were used. Although the results of the proposed method were improved by using the GA tool, the consequence was higher process complexity and a longer processing time according to the specified scope and conditions. An optimal parameter search should remain in a limited area to avoid searching unnecessary areas.

An effective prediction method is still essential for reducing distortion and size of the location maps. In the future, dynamic predictors for each image and each embedding size should be considered.

Table 1 Parameter values and PSNR values obtained from GA tested with different payload sizes for V1, V2 and V14 images.

Pay Load	QR Code V1	QR Code V2	QR Code V14
	***W*, *S*, PSNR**	***W*, *S*, PSNR**	***W*, *S*, PSNR**
10k	0.980, 6, 58.56	0.972, 7, 59.87	0.980, 27,67.662
40k	---	0.998, 2, 53.27	0.988, 18, 61.40
100k	---	---	0.976, 2, 57.26
300k	---	---	---
700k	---	---	---
1000k	---	---	---

Table 2 Parameter values and PSNR values obtained from GA tested with different payload sizes for V22, V31 and V40 images.

Pay Load	QR Code V22	QR Code V31	QR Code V40
	W, *S*, PSNR	*W*, *S*, PSNR	*W*, *S*, PSNR
10k	0.972, 18,70.77	0.972, 15,72.79	0.972, 10, 75.04
40k	0.980, 13, 64.30	0.972, 14,66.69	0.984, 10, 68.60
100k	0.980, 4, 60.29	0.984, 4, 62.70	0.972, 8, 64.59
300k	0.992, 3, 55.16	0.992, 2, 57.87	0.972, 4, 59.80
700k	---	0.992, 1, 53.48	0.972, 3, 55.83
1000k	---	---	0.976, 1, 53.87

4. CONCLUSION

A survey of the appropriate parameters of the PEHS + sorting RDH algorithm using GA and LMSW strategy in limited scope for each QR code image and each payload is presented in order to achieve the best possible results. Different payload sizes were used to test the performance of the proposed method and the previous methods. The experimental results showed that the proposed GA was superior compared to the two previous similar approaches. However, the average embedding ratio efficiency of the proposed method is still only half the image size, which should be improved. In the future, multi-bit per pixel embedding strategies for smooth regions should be exploited to embed more information while maintaining low distortion.

6. ACKNOWLEDGMENT

This research was supported by the Faculty of Engineering, Rajamangala University of Technology Rattanakosin (RMUTR).

7. REFERENCES

Mintzer, F., Lotspiech, J. and Morimoto, N. (1997). Safeguarding digital library contents and users: digital watermarking, *D-Lib Magazine*., 3, 33-45.

Fridrich, J. Goljan, M. and Du, R. (2002). Lossless data embedding for all image formats. *Proceedings of the SPIE*., 4675, (pp. 572-583).

Celik, M. Sharma, U. G. Tekalp, A. M. and Saber, E. (2002). Reversible data hiding. *Proceedings of the IEEE International Conference on Image Processing*, 2, (pp. II/157-II/160).

Yang, B. Schmucker, M. Funk, W. Busch, C. and Sun, S. (2004). Integer DCT-based reversible watermarking for images using companding technique. *Proceedings of SPIE-The International Society for Optical Engineering*, 5306, (pp. 405-415).

Xuan, G. Chen, J. Zhu, J. Shi, Y. Q. Ni, Z. and Su, W. (2002). Lossless data hiding based on integer wavelet transform. *Proceedings of the IEEE Workshop on Multimedia Signal Processing*, (pp. 312-315).

Xuan, G. Shi, Y. Q. Ni, Z. C. Chen, J. Yang, C. Zhen, Y. and Zheng, J. (2004). High capacity lossless data hiding based on integer wavelet transform. *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2, (pp. II29-II32).

Xuan, G. Yang, C. Zhen, Y. Shi, Y. Q. and Ni, Z. (2004). Reversible data hiding based on wavelet spread spectrum. *Proceedings of the IEEE 6th Workshop on Multimedia Signal Processing*, (pp. 211-214).

Zou, D. Shi, Y. Q. Ni, Z. and Su, W. (2006). A semi-fragile lossless digital watermarking scheme based on integer wavelet transform. *IEEE Trans. Circuits Syst. Video Technol*., *16*(10), 1294-1300.

Tian, J. (2003). Reversible Data Embedding Using a Difference Expansion. *IEEE Trans. Circuits Syst. Video Technol*., *13*(8), 890-896.

Kamstra, L. H. J. and Heijmans, A. M. (2005). Reversible Data Embedding into Images Using Wavelet Techniques and Sorting. *IEEE Trans. Image Process*., *14*(12), 2082-2090.

Kim, H. J. Sachnev, V. Shi, Y. Nam, Q. J. and Choo, H. G. (2008). A novel difference expansion transform for reversible data embedding. *IEEE Trans. Inf. Forensics Secur*., *3*(3), 456-465.

Ni, Z. Shi, Y. Q. Ansari, N. and Su, W. (2006). Reversible data hiding. *IEEE Trans. Circuits Syst. Video Technol*., *16*(3), 354-362.

Thodi, D. M. and Rodriguez, J. J. (2007). Expansion Embedding Techniques for Reversible Watermarking. *IEEE Trans, Image Process*., *16*(3), 721-730.

Sachnev, V. Kim, H. J. Nam, J. Suresh, S. and Shi, Y. Q. (2009). Reversible Watermarking Algorithm Using Sorting and Prediction. *IEEE Trans. Circuits Syst. Video Technol*., *19*(7), 989-999.

Kotvicha, A. Sanguansat, P. and Kasemsa, M. L. K. (2012). Expand Variance Mean Sorting for Reversible Watermarking. *Int. J. Comput. Commun. Eng*., *1*, 196-199.

Li, X., Li, B., Yang, B., & Zeng, T. (2013). General framework to histogram-shifting-based reversible data hiding. *IEEE Transactions on Image Processing*, *22*, 2181–2191.

Ou, B., Li, X., Zhao, Y., & Ni, R. (2014). Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion. *Signal Processing: Image Communication*, *29*, (pp. 760–772).

Peng, F., Li, X., & Yang, B. (2014). Improved PVO based reversible data hiding. *Digital Signal Processing*, *25*, 255–265.

Wang, X., Ding, J., & Pei, Q. (2015). A novel reversible image data hiding scheme based on pixel value ordering and dynamic pixel block partition. *Information Sciences*, *310*, 16–35.

Qu, X., & Kim, H. J. (2015). Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding. *Signal Processing*, *111*, 249–260.

Ou, B., Li, X., & Wang, J. (2016). 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.

Dragoi, I. C., Caciula, I., & Coltuc, D. (2018). Improved pairwise pixel-value-ordering for high-fidelity reversible data hiding. *In IEEE International Conference on Image Processing,* (pp. 1668–1672).

Gao, E., Pan, Z., & Gao, X. (2019). Reversible data hiding based on novel pairwise PVO and annular merging strategy. *Information Sciences*, *505*, 549–561.

He, W., & Cai, Z. (2020). An insight into pixel value ordering prediction-based prediction-error expansion. *IEEE Transactions on Information Forensics and Security*, *15*, 3859–3871.

Zhang, T., Li, X., Qi, W., & Guo, Z. (2020). Location-based pvo and adaptive pairwise modification for efficient reversible data hiding. *IEEE Transactions on Information Forensics and Security*, *15*, 2306–2319.

Fan, G., Pan, Z., Gao, E., Gao, X., & Zhang, X. (2021). Reversible data hiding method based on combining IPVO with bias-added Rhombus predictor by multi-predictor mechanism. *Signal Processing*, 180, Article 107888.

Panyindee, C. (2021). Efficient Reversible Data Hiding Using Dynamic Variance Mean Sorting and Fitting Weight Rhombus Predictor. *Journal of Circuits, Systems, and Computers*, *30*(9), 2150169-1-2150169-17.

Fan, G., Pan, Z., Zhou, Q., & Zhang, X. (2023). Flexible patch moving modes for pixel-value-ordering based reversible data hiding methods. *Expert Systems with Applications*, *214*, 1–14.

Dangmee, P. and Lilakiatsakun, W. (2015). Steganography hiding data within QR Code. *J. Inf. Sci. Technol*., *5*(1), 35-41.

Panyindee, C. Leelawiwat, T. and Rangsirak, P. (2019). Study and Development of a Reversible Data Hiding Algorithm Using Linear Weighting for QR Code images. *Engng. J. CMU*., *26*(1), 80-92.

Panyindee, C. Leelawiwat, T. and Rangsirak, P. (2019). Improved Sorting in QR Code images Using EVM technique for Reversible Data Hiding. *Engng. J. CMU*., *26*(3), 66-76.

Hu, Y. Lee, H. K. and Li, J. (2009). De-based reversible data hiding with improved overflow location map. IEEE Trans. *Circuits Syst. Video Technol., 19*(2), 250-260.

Li, X. Yang, B. and Zeng, T. (2011). Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. *IEEE Trans. Image Process*., *20*(12), 3524-3533.

Kang, S. Hwang, H. J. and Kim, H. J. (2012). Reversible watermark using an accurate predictor and sorter based on payload balancing. *ETRI*., *34*(3), 410-420.

Ou, B. Li, X. Zhao, Y. and Ni, R. (2013). Reversible data hiding based on PDE predictor. *J. Syst. Softw*., *86*(10), 2700-2709.

Panyindee, C. and Pintavirooj, C. (2016). Optimal Gaussian Weight Predictor and Sorting Using Genetic Algorithm for Reversible Watermarking Based on PEE and HS. *IEICE Trans. Info Syst*., *E99-D* (9), 2306-2319.

Project Nayuki. (2021). *QR Code generator library*. https://www.nayuki.io/page/qr-code-generator-library.