http://ansinet.com/itj

ITJ

ISSN 1812-5638

# INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 13 (15): 2374-2384, 2014
ISSN 1812-5638 / DOI: 10.3923/itj.2014.2374.2384


# A Reversible Data Hiding Method Using Improved Neighbor Mean Interpolation and Random-Block Division

[1]Ling Liu, [2]Tungshou Chen, [1]Shunzhi Zhu, [1]Weidong Hong and [3]Xin Si
[1]School of Computer and Information Engineering,
Xiamen University of Technology, Xiamen, 361024, China
[2]Faculty of Computer Science and Information Engineering,
National Taichung University of Science and Technology, Taichung, Taiwan
[3]Department of Mathematics, Xiamen University of Technology, Xiamen, 361024, China

**Abstract:** This study proposes a novel data-hiding method based on an improved neighbor mean interpolation and random-block division. The basic idea of the proposed method is to use the proposed neighbor mean interpolation as the scaling-up strategy and to make full use of the correlation between neighboring pixels to increase the payload without sacrificing much distortion for data hiding. Moreover, the random-block division technique is adopted to increase the security of data hiding method under the detection of a statistical steganalysis tool. Experimental results showed that our proposed method can obtain a large embedding capacity while maintaining high visual quality image and undetectability. Compared with another similar work, the PSNR is guaranteed to be higher than 34 dB which maintained a maximum capacity increase of 22.5-36.5%. Also, the undetectabilty can be improved by a maximum of 33.66%.

**Key words:** Reversible data hiding, interpolation, random-block division, scaling-up, undetectability

## INTRODUCTION

Data hiding is a technique which embeds secret information into a carrier for transferring information confidentially (Provos and Honeyman, 2003). The digital images often served as carriers in the information transmission process. The image for carrying secret data is called the 'cover image' and the image carrying secret data is called the 'stego image'. After data embedding, pixel values of the cover image will be changed and thus the image quality is degraded. In order to evade statistical detection the distortion caused by secret data embedding should be as small as possible. In general, embedding capacity, visual quality and undetectability are important evaluation parameters to measure the performance of data hiding (Wang and Wang, 2004).

Many data hiding techniques cannot recover the stego image to its original state after extracting the embedded message. Also, the distortions caused by data embedding are permanent (Zhang and Wang, 2006; Guo, 2007). However, for some special applications such as medical or military images, where the requirements for data hiding is higher, the stego image must maintain less distortion and the original cover image can be completely recovered after extracting the secret data. Therefore, reversible data hiding algorithms have been extensively developed where the original image can be recovered after extracting the embedded data.

Reversible data hiding can be grouped into four categories, difference expansion, histogram shifting, prediction and interpolation (Yang *et al*., 2012). The main idea of the difference expansion methods (Tian, 2003; Alattar, 2004; Hsiao *et al*., 2009) is to use the difference values of a pair of pixels from the cover image and expand the difference values to embed secret message. These methods have higher payloads but undesirable distortion due to the expand differences of pixel pairs. Most histogram shifting methods (Ni *et al*., 2006; Hwang *et al*., 2006; Lin *et al*., 2008) employ the histogram of the cover image for hiding secret data. They are simple and effective. However, the capacity is restricted for most applications. Many researchers tried various approaches to maximize the capacity of histogram shifting methods. For example, Kim *et al*. (2009) proposed a reversible data hiding method that exploits high spatial correlation among sub-sampled images. The method raised the embedding capacity of histogram shifting by using sub-sampling techniques. Some variants of Kim's work can be found in (Luo *et al*., 2011; Liu *et al*., 2013). Prediction-based methods (Tsai *et al*., 2009; Hong and Chen, 2010;

**Corresponding Author:** Shunzhi Zhu, School of Computer and Information Engineering, Xiamen University of Technology, Xiamen, 361024, China

Sachnev *et al*., 2009) made use of neighboring pixels or pixel blocks to predict pixel values in the target image for hiding secret message. For example, Tsai *et al*. (2009) proposed a reversible data hiding method based on predictive coding and histogram shifting which obtains better embedding capacity by exploiting the similarity of neighboring pixels to construct a histogram of prediction errors. Hong and Chen (2010) extended Tsai's method and used a set of basic pixels to improve the prediction accuracy and thus improving the embedding capacity. Furthermore, Sachnev *et al*. (2009) proposed a diamond prediction scheme to improve the prediction accuracy and obtained better payload.

Among data hiding methods based on interpolation, (Jung and Yoo, 2009) proposed a data hiding method by using neighbor mean interpolation. This method employs a novel interpolation method to enlarge the input image. The scaled-up image servers as the cover image which is used in the data embedding process. The proposed scaling-up interpolation algorithm provides the advantages of both a low-time complexity and high computing speed. Luo *et al*. (2010) proposed a reversible data hiding scheme based on additive interpolation-error expansion. This method applies an interpolation technique to generate residual values which we often called interpolation-errors which are then expanded by addition to embed secret data. The strategy is efficient and thus obtaining very low distortion and relatively large capacity. However, Jung *et al*.'s method does not fully exploit the correlation of neighboring pixels in an image, leading to reduce amount of payload. Besides, Jung *et al*. did not consider the security of the data hiding method. In this study, we modified Jung *et al*.'s method so that more correlation among neighboring pixels could be used for increasing the embedding capacity. Furthermore, the random-block division technique is adopted to increase the security of data hiding method under the detection of statistical steganalysis tool. Compared with another similar study, the embedding capacity can be significantly increased and security is improved while assuring a good visual quality.

## INTERPOLATION METHOD

**Jung *et al*.'s method:** Jung *et al*. proposed a novel interpolation method in 2009. This method uses neighboring pixel values in a cover image to obtain the mean which is inserted into a pixel to be allocated in a scaled-up image. This method obtained less distortion and greater image resolution.

Figure 1 shows an example of Jung *et al*.'s method. Suppose p(i, j) and p'(i, j) are denoted pixel values in the cover image Fig. 1a and the scaled-up image Fig. 1c, respectively. By using Jung *et al*.'s method, we can get the results p'(0, 0) = p(0, 0), p'(0,2 ) = p(0, 2), p'(2, 0) = p(2, 0), p'(0, 1) = (p'(0, 0)+p'(0, 2))/2, p'(1, 0) = (p'(0, 0)+p'(2, 0))/2 and p'(1, 1) = (p'(0, 0)+ p'(0, 1)+p'(1, 0))/3.

**Proposed interpolation method:** As we can see from Fig. 1, the pixel value p'(1, 1) is only indicated by p(0, 0), p(0, 2) and p(2, 0). The pixel p(2, 2) does not serve as a reference pixel to calculate p'(1, 1). Moreover, p'(0, 1) and p'(1, 0) are calculated only by two neighboring pixel values and ignoring the other pixels in the scaled-up image. These limit the embedding capacity or degrade image resolution. In this study, we propose a novel improved neighbor mean interpolation method. In the scaling-up process, pixels scaled-up are calculated by the full use of four pixels in the cover image. In this way, more neighboring pixel values are used to calculate the value to be allocated and therefore we can get higher resolution pixels. The detailed process is:

- Suppose the size of a cover image is M×N pixels. After generating the scaling-up process, the scaled-up image is sized (2M-1)×(2N-1) pixels. The scaling-up process is similar to that of Jung *et al*.'s method in which four-pixel, 2×2 pixel blocks as a scaling-up unit is enlarged and scaled-up unit of size 3×3 is obtained. Suppose pixel $p'_m(i_m, j_m)$ represents the center pixel in the scaled-up unit, as shown in Fig. 1c which can be computed in Eq. 1:

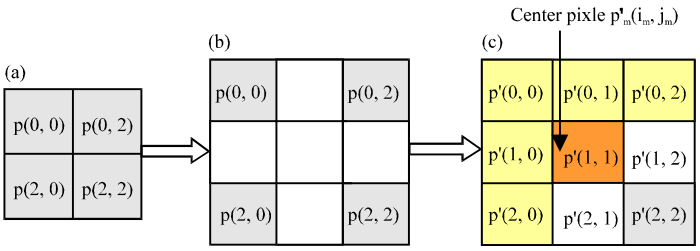


Fig. 1(a-c): An example of the scaling-up process on Jung *et al*.'s method (a) Cover image (b) Scaling-up process and (c) Scaled-up image

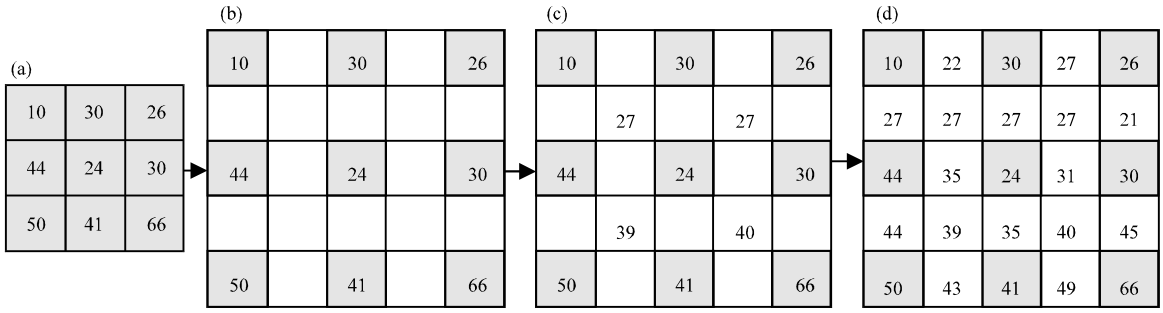


Fig. 2(a-d): An example of the scale-up process (a) A cover image of sized 3×3, (b) Insert blank rows and columns into the cover image, (c) Values of the center pixels and (d) Scaled-up image

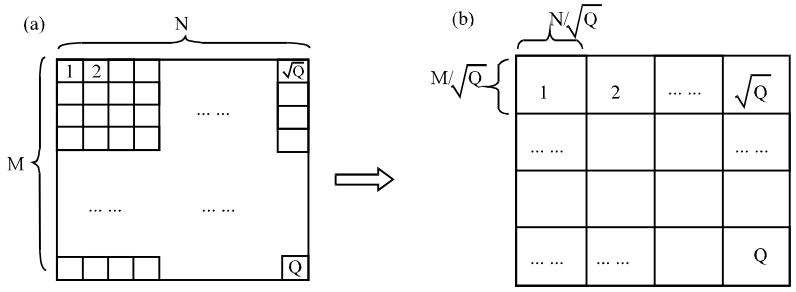


Fig. 3(a-b): Image segmentation (a) Original image and (b) Image blocks

$$p'_m(i_m, j_m) = \frac{\lfloor p'(i_m-1, j_m-1) + p'(i_m-1, j_m+1) + p'(i_m+1, j_m-1) + p'(i_m+1, j_m+1) \rfloor}{4} \quad (1)$$

- By Eq. 2, compute other pixel values in the scaled-up unit expected for the center pixel $p'_m(i_m, j_m)$:

$$p'(i, j) = \begin{cases} (p'(i, j-1) + p'(i, j+1) + p'(i+1, j)) / 3 & \text{if } i = i_m - 1,\ j = j_m \\ (p'(i-1, j) + p'(i+1, j) + p'(i, j+1)) / 3 & \text{if } i = i_m,\ j = j_m - 1 \\ (p'(i, j-1) + p'(i, j+1) + p'(i-1, j)) / 3 & \text{if } i = 2(M-1) \\ (p'(i, j-1) + p'(i, j+1) + p'(i-1, j)) / 3 & \text{if } j = 2(N-1) \end{cases} \quad (2)$$

Figure 2 describes the scaling-up procedure of the cover image sized 3×3. First, insert blank rows and blank columns between two adjacent rows and two adjacent columns, respectively, as shown in Fig. 2b. Then, by using Eq. 1, the results of the center pixel value are 27, 27, 39 and 40, respectively, as shown in Fig. 2c. By Eq. 2, other pixel values are also computed. For pixel-block (10, 30, 44, 24), the scaled-up pixel values are 22 and 27. Repeat the above scaling-up process until no more pixel-block will be scaled-up and the final scaled-up image is sized 5×5 pixels, as shown in Fig. 2d.

## PROPOSE DATA HIDING METHOD

Here, we propose a data hiding method based on improved mean interpolation and random-block division. Before embedding secret data, the cover image is first enlarged by the proposed interpolation which resulted in a scaled-up image then the scaled-up image is partitioned into Q blocks and finally, a pseudo-random sequence is applied to mark and reorder the scaled-up image blocks. The sequence of data hiding can thereby, be in the reordered-blocks direction. In data embedding, a method similar to that of Jung *et al.*'s method is used. After data extraction, the cover image can be completely recovered.

**Division based on random-block selecting:** The original image sized M×N is partitioned into Q blocks. Each block is composed of $M/\sqrt{Q} \times N/\sqrt{Q}$ pixels. Figure 3 gives an example of the division. Sequentially scan the blocks and get result 1, 2, 3,..., $\sqrt{Q}$,..., Q. A pseudo random sequence $\{a_1, a_2, \ldots, a_Q\}$ is used to reorder the blocks, where $1 \le a_i \le Q$, $1 \le i \le Q$. Repeat scanning of the blocks and the results are $a_1, a_2, \ldots, a_Q$. After that, the order of data hiding will be in zig-zag direction.

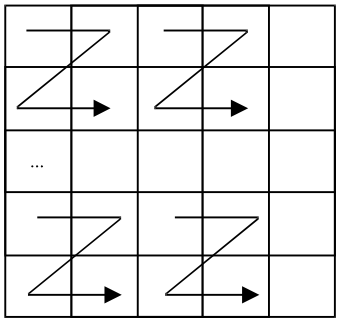


Fig. 4: Zig-zag scanning

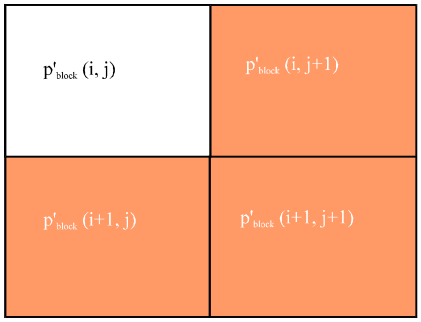


Fig. 5: Domain (shaded)

**Embedding procedure**

**Input:** Cover image I, secret bit string w and pseudo-random sequence.

**Output:** Stego image $I'_w$ and overhead information $O_{info}$

The data embedding steps are as follow:

**Step 1:** Enlarge the cover image I using our proposed method and obtain a scaled-up image I'

**Step 2:** Partition the scaled-up image I' into Q blocks and mark the position of Q blocks with pseudo-random sequence to determine the sequence of data embedding

**Step 3:** For each block, perform data embedding. Detailed steps are listed as follow:

(a) Before embedding secret data, we first partition each block into four-pixel, non-overlapping units by zig-zag scanning, as shown in Fig. 4. For each unit repeat step (b-c) until no unit needs to be embedded

(b) Figure 5 shows a 2×2 pixel unit where the pixel values are denoted by $p'_{block}(i,j)$, $p'_{block}(i, j+1)$, $p'_{block}(i+1, j)$ and $p'_{block}(i+1, j+1)$. The secret data is embedded into three pixels except for $p'_{block}(i, j)$. Suppose the domain to be embedded is denoted by Δ domain (as shown in shaded boxes in Fig. 5). The pixel values in Δ domain are denoted by $p'_{block}(i_\Delta, j_\Delta)$ and the payload of each pixel in Δ domain is in Eq. 3:

$$d = p'_{block}(i_\Delta, j_\Delta) - p'_{block}(i, j), \text{ where, } (i_\Delta, j_\Delta) \in \Delta \quad (3)$$

(c) Compute the number of embedded bits, n, in every pixel in Δ domain using Jung *et al.*'s method. Then obtain n bits secret data and convert it to integer value b. Finally, calculate the stego pixel $p'_w(i_w, j_w)$ by using integer value b as in Eq. 4:

$$p'_w(i_w, j_w) = p'_{block}(i_\Delta, j_\Delta) + b \quad (4)$$

**Step 4:** The result is stego image $I'_w$ with embedded secret by blocks. The pseudo-random sequence and Q are outputted as overhead information $O_{info}$ for decoding

**Extraction and image recovery procedures:** Before extracting the hidden secret data, the pseudo-random sequence and Q are extracted from the overhead information. The detailed procedure is as follows.

**Input:** Stego image $I'_w$ and overhead information $O_{info}$.

**Output:** Cover image I and secret data w. The extraction and image recovery steps are as follows:

**Step 1:** Partition the stego image $I'_w$ into Q blocks

**Step 2:** For each stego block, repeat step (d) and (g) until all secret data are extracted. The sequence for extracting data from Q blocks is in the order of pseudo-random sequence

(d) Partition stego block into nine-pixel, overlapping and consecutive stego units by zig-zag scanning as shown in Fig. 6

(e) For each stego unit (Eq. 1-2), compute the values of center pixel and other scaled-up pixels in original scaled-up image I', respectively; pixel values $p'_{block}(i_\Delta, j_\Delta)$ in Δ domain are computed

(f) Before extracting secret data, we first partitioned each stego block into four-pixel, non-overlapping units by zig-zag scanning as shown in Fig. 4

(g) For each unit, mark the pixels in Δ domain and compute the payload of each pixel in Δ domain by using Eq. 3. Then obtain the number of bits n embedded in Δ domain and secret data $b_w$ can be calculated by using Eq. 5

$$b_w = p'_w(i_w, j_w) - p'_{block}(i_\Delta, j_\Delta), \text{ where } (i_w, j_w) \in \Delta \quad (5)$$

Finally, integer value $b_w$ is converted to secret data with n bits, repeat g) process until all secret data is extracted. The final extracted secret is w

**Step 3:** After extracting all secret data, the scaled-up image I' can be obtained by grouping Q blocks. In the scaled-up image I', first and last rows remain unchanged, first and last columns remain unchanged and delete the scaled-up rows and columns to get the cover image I

**An example of the proposed method:** Here, an example is used to illustrate Jung *et al.*'s method in order to analyze our proposed interpolation method. Assume that cover image pixels to be 46, 112, 210 and 90, as shown in Fig. 7a. By Eq. 1 and 2, pixel values in the scaled-up images are 90, 123, 114 and 105 as shown in Fig. 7b. Suppose secret data is $(1001101011000101 0)_2$. The 2×2 four-pixel unit is (46, 90, 123, 114). By Eq. 3, the difference values are 44, 77 and 68, respectively. Then similar to Jung *et al.*'s method calculate the number of bits embedded in Δ domain. The results are 5, 6 and 6, respectively. A secret data with n bits in the embedding data is used and converted to integer value b. The results are $(10011)_2 = 19$, $(010110)_2 = 22$, $(001010)_2 = 10$, respectively. Finally, stego pixel values are obtained by Eq. 4 and the stego image is shown as Fig. 7c. The data extraction and image recovery can be completed by the inverse data embedding. After receiving the stego image (Fig. 7c), the pixel values in the scaled-up images are first calculated by using Eq. 1-2, the results are 114, 90, 123, 105 and 138, respectively, the scaled-up image is shown as Fig. 7d. By Eq. 3, the maximal embedding capacity of each pixel in Δ domain are 44, 77 and 68, respectively. Consequently, the numbers of bits embedded are 5, 6 and 6, respectively. Then, by Eq. 5, secret data embedded are 19, 22 and 10, respectively. The secret data in the form of integer value is converted to data with n bits, i.e., the results are $19 = (10011)_2$, $22 = (010110)_2$, $10 = (001010)_2$. The extracted data is 100110101 1 0 0 01010 and the recovery image is shown as Fig. 7e.

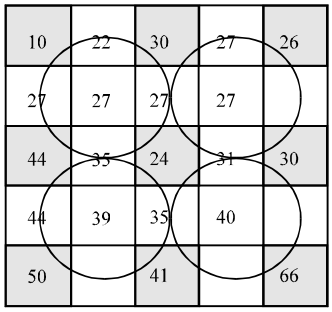


Fig. 6: Partitioning stego block

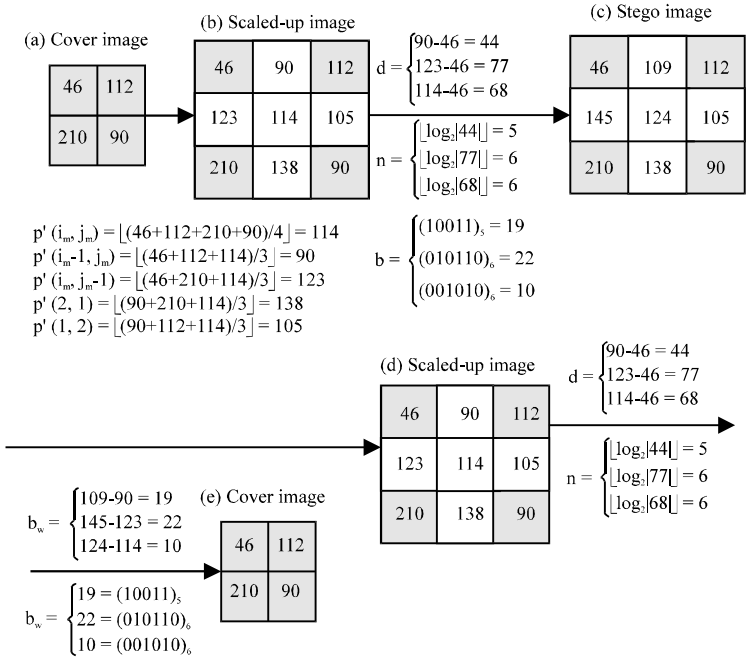


Fig. 7(a-e): An example for the proposed method, 1: Embedding procedure (a-c) and 2: Data extraction and image recovery process (d-e)

## EXPERIMENTAL RESULTS AND ANALYSIS

Embedding capacity, visual quality and undetectability are used for performance evaluations in the data hiding algorithm. In experimental results and analysis MATLAB was adopted to carry out the experiments and the above three parameters were analyzed by comparing with Jung *et al.*'s method.

**Embedding capacity and visual quality:** In the data embedding, pixel values modification often occurs so the image quality will be degraded. The Peak Signal-to-Noise Rate (PSNR) was used to measure image visual quality:

$$PSNR = 10\log\left(\frac{255^2}{MSE}\right) \quad (6)$$

where, MSE is the mean square error between the scaled-up image and stego image. MSE is defined as:

$$MSE = \frac{1}{M\times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left(I'_w(i,j) - I'(i,j)\right)^2 \quad (7)$$

where, M×N denotes the scaled-up image size, I' (i, j) and $I'_w(i,j)$ denote the pixel values of the scaled-up image and the stego image, respectively.

In our experiment, nine images sized 512×512 from USC-SIPI (http://sipi.usc.edu/services/database/database.cgi?volume=misc) image database were taken as test images as shown in Fig. 8. The test images were scaled-down and then generated the scaled-up images using the proposed interpolation methods. The scaled-up


Fig. 8(a-i): Test images (a) Airfield, (b) Baboon, (c) Barbara, (d) Boat, (e) Couple, (f) Crowd, (g) Lena, (h) Peppers and (i) Airplane

images were used for embedding secret data. In here, nine 512×512 test images were scaled-down to generate 256×256 cover images which were used as input source of the proposed interpolation method. Consequently, by using the proposed method, 512×512 scaled-up images were obtained for embedding secret information. The PSNR value and capacity are measures of the visual quality and payload of the interpolation and hidden data results.

**Undetectability:** A good data hiding method should be able to evade statistical detection. Only then could senders transmit confidential information to receivers securely. In this study, the undetectability of the proposed method under Subtractive Pixel Adjacency Matrix (Pevny *et al*., 2010) (SPAM) was analyzed. SPAM is a good steganalysis technique of detecting stego images and it gets the features of images by computing the transition probabilities along eight directions. We use a soft-margin Support Vector Machine (SVM) with Gaussian kernel to implement the SPAM steganalyzer. The error rate can be calculated in Eq. 8:

$$P_{Err} = \frac{1}{2}(P_{Fp} + P_{Fn}) \quad (8)$$

The calculated result is used to evaluate the undetectability of a data-hiding method under SPAM. Here, $P_{Fp}$ and $P_{Fn}$ are the false alarm rate and false denying rate, respectively. The higher the error rates of data hiding method against SPAM detection, the better the undetectability.

The undetectability of the proposed method using SPAM is analyzed by comparing with Jung *et al*.'s method. All test images were obtained from the RSP image database online. RSP consists of 10,000 gray-scale images of size 512×512. We selected 5,000 images from RSP image database. Half of the images were used for data hiding and the other half for detection. Different amount of payload, varying from 0.01-0.05 bpp, was embedded to measure the undetectability of the data hiding method. We obtained SPAM features from literature (http://dde.binghamton.edu/download/spam/), calculated the classification error by adopting five-fold cross-validation and found the penalization parameter C and the kernel parameter γ by means of the Simulated Annealing (SA) optimization.

## Experimental results

**Parameter value Q analysis:** Figure 9 shows the maximal capacity of the proposed method at various Q values (block numbers). As shown in Fig. 9: In case of Q≤9; the smaller Q is the greater the capacity: In the case of Q≥16, the greater Q is the smaller the capacity. This is due to the reason that a higher Q value results in more partition which causes pixel loss.

Figure 10 shows the PSNR values of the proposed method with maximum payload at various Q values. It can be seen that the visual quality is the best in the case of Q = 9 which is owing to the least embedding data; If Q is between 16 and 256, visual quality vary inconspicuously; If Q>256, visual quality becomes higher due to more block numbers and the least secret data will be embedded.

Figure 11 shows the error rate obtained by SPAM for the proposed method at various Q values. It can be seen that the error rate is between 19.42 and 22.56%. When block number Q equals to 16, the error rate of the proposed method using SPAM can reach the maximum of about 22.56% with higher security.

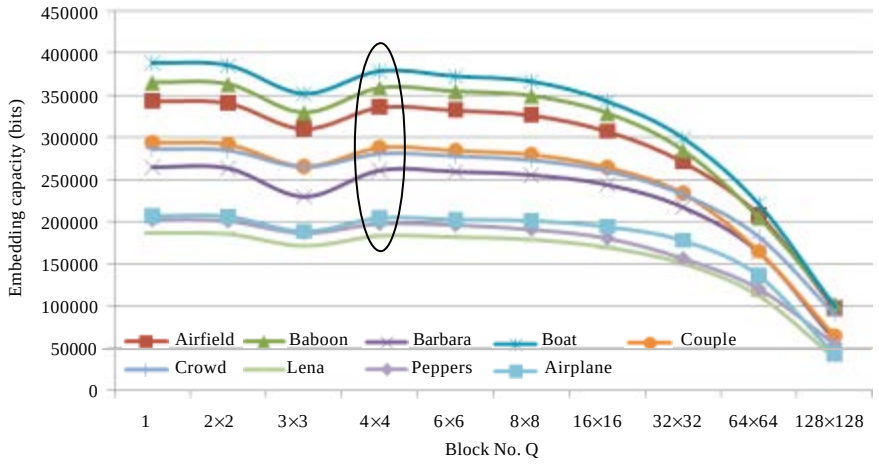


Fig. 9: Maximal capacity of the proposed method at various Q values

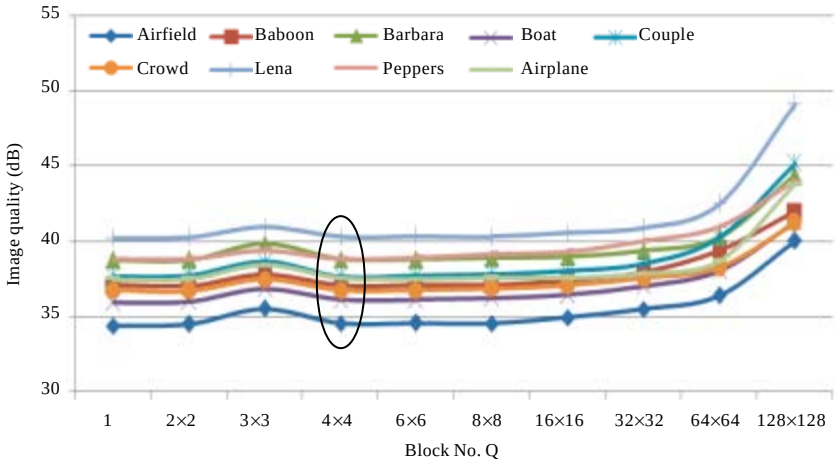


Fig. 10: PSNR of stego image for the proposed method with maximum payload at various Q values

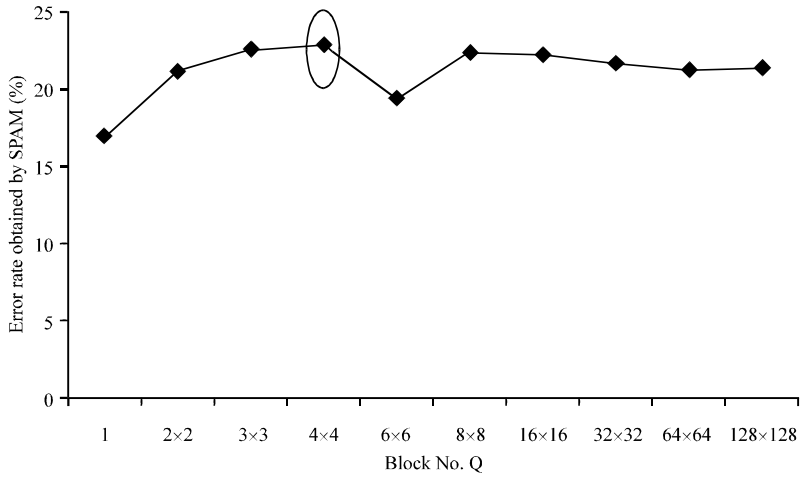


Fig. 11: Error rate obtained by SPAM for the proposed method at various Q values ($T^1 = 4$)

As in (Fig. 9-11), embedding capacity, visual quality and undetectability of the proposed method are significantly high and block number Q = 16 will be used in the proposed method.

**Experimental results (embedding capacity and visual quality):** Figure 12 shows the visual impacts of marked images obtained by the proposed method at various embedding capacities upto 0.5 bpp. It indicates that higher embedding capacity lowers the visual quality of the proposed method. In our experiments, 94.4% of the PSNR values of the proposed method are over 40 dBs. This means the proposed method shows significantly high visual quality.

Table 1 shows comparison of the maximum capacity and PSNR for the proposed method and Jung *et al.*'s method. It can be seen that the maximum capacities of the proposed method are significantly higher than Jung *et al.*'s method. For example, in test image Airfield, the maximum capacity of the proposed method is 335914 bits, whereas Jung *et al.*'s method is 271505 bits. For other test images, the proposed method also performed better than Jung *et al.*'s method. PSNR of the proposed method is slightly lower than that of Jung *et al.*'s. It is due to when payload increases, PSNR value decreases. Besides, block-segmentation also causes distortion. But PSNR values of the proposed method with maximum payload is over 34 dB which showed comparable visual quality.

**Experimental results (undetectability):** Figure 13 shows the error rate of the proposed method against SPAM

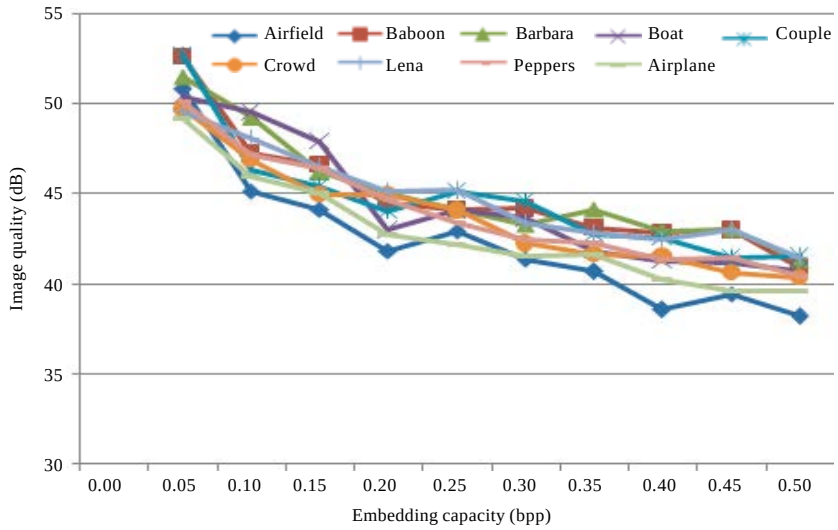


Fig. 12: Comparison of embedding capacity (bpp) vs. distortion (dB)

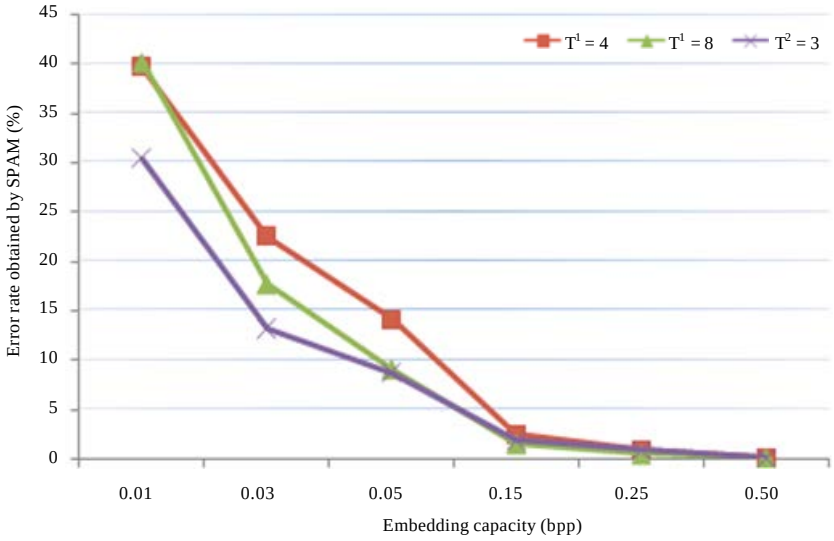


Fig. 13: Error rate obtained by SPAM for the proposed method

Table 1: Comparison of PSNR and the maximum capacity

| | Jung and Yoo (2009) | | Proposed method | |
|---|---|---|---|---|
| Image | PSNR | Capacity | PSNR | Capacity |
| Airfield | 36.88 | 271505 | 34.53 | 335914 |
| Baboon | 39.45 | 292654 | 37.06 | 358492 |
| Barbara | 41.34 | 204568 | 38.80 | 261381 |
| Boat | 38.50 | 306306 | 36.11 | 378543 |
| Couple | 40.28 | 210941 | 37.63 | 287928 |
| Crowd | 39.36 | 221595 | 36.72 | 280556 |
| Lena | 42.95 | 135594 | 40.29 | 183595 |
| Peppers | 41.58 | 147638 | 38.83 | 197903 |
| Airplane | 40.03 | 153601 | 37.52 | 204952 |

detection at various embedding capacities upto 0.5 bpp. As a result, regardless of T value, higher embedding capacities resulted in weaker undetectability of the proposed method against the detection of SPAM. Furthermore, the error rate is the lowest in the case of $T^2 = 3$. This is due to the reason that increasing the order of the Markov chain proved to be highly beneficial as the accuracy of the resulting steganalyzers has significantly increased, making the detectability of SPAM steganalyzer better and thus weakening the undetectability.

Figure 14a-c showed the comparison results of undetectability under SPAM steganalyzer for the proposed method and Jung *et al.*'s method at various payloads upto 0.5 bpp. As can be seen from Fig. 14a-c, regardless of T value, the error rates obtained by our

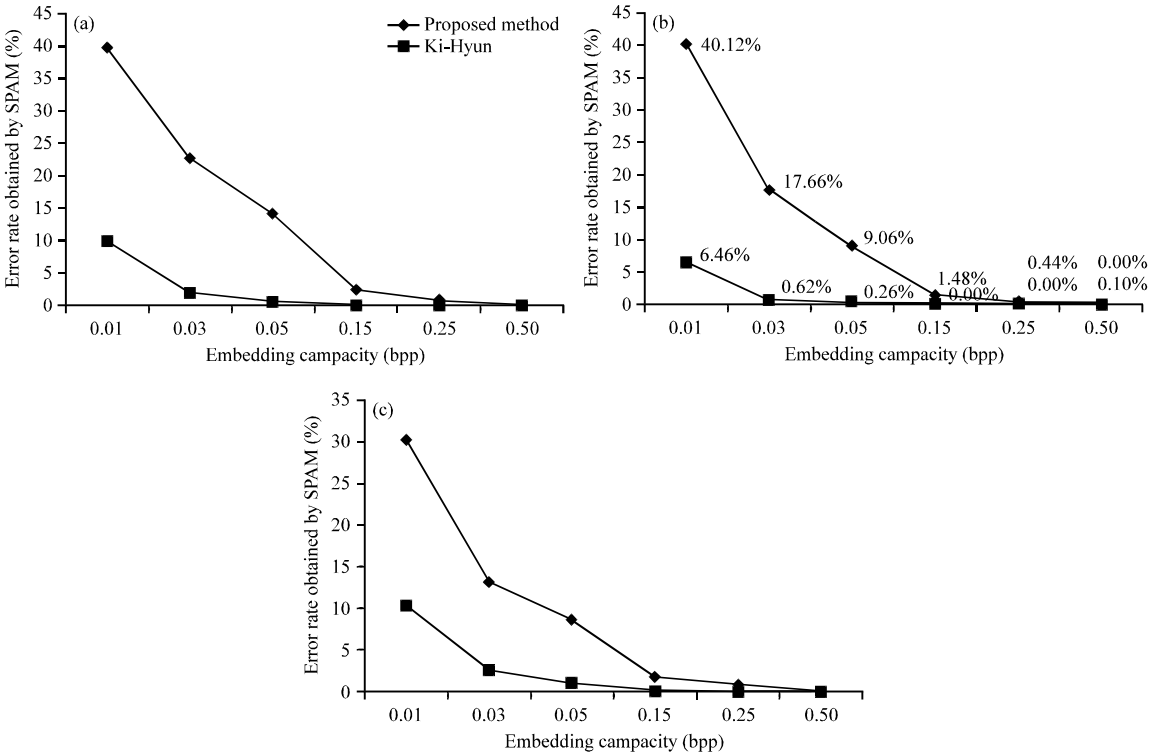


Fig. 14(a-c): Comparison of error rate obtained by SPAM for the proposed method and Jung *et al.*'s method (a) $T^1 = 4$, (b) $T^1 = 8$ and (c) $T^2 = 3$

proposed method are significantly higher than those obtained by Jung *et al.*'s method. For example, for embedding capacity 0.01 bpp and $T^1 = 8$ while the error rate of the proposed method using SPAM is 40.12 and 6.46% for Jung *et al.*'s method. That is, the gain in the undetectability is 33.66% at 0.01 bpp. All experimental results agreed with the fact that our proposed method is more secure under the detection of SPAM steganalyzer than Jung *et al.*'s method.

## CONCLUSION

This study proposed a simple and effective reversible data hiding algorithm by using improved neighbor mean interpolation and random-block selecting. The proposed interpolation method employs every four pixels in the cover image as reference pixels for scaling-up. This makes the scaled-up images symmetrical and smoother. Thus it allows embedding a large amount of secret data while keeping a very high visual quality. During the data embedding process, random pixel-block was selected to be used to embed information, making the distribution of the information embedded to be irregular and thus enhancing the security of the proposed method. Our experimental results show that the maximum capacity and undetectability of the proposed method can be improved by 36.5 and 33.66% to the maximum extent.

## ACKNOWLEDGMENTS

This work is partially supported by the National Natural Science Foundation of China (Grant No. 61373147) and the Science and Technology Research Projects of XiaMen University of Technology (Grant No. YKJ11012R, YKT10037R).

## REFERENCES

Alattar, A.M., 2004. Reversible watermark using the difference expansion of a generalized integer transform. IEEE Trans. Image Process., 13: 1147-1156.

Guo, J.M., 2007. Improved data hiding in halftone images with cooperating pair toggling human visual system. Int. J. Imag. Syst. Technol., 17: 328-332.

Hong, W. and T.S. Chen, 2010. A local variance-controlled reversible data hiding method using prediction and histogram-shifting. J. Syst. Software, 83: 2653-2663.

Hsiao, J.Y., K.F. Chan and J.M. Chang, 2009. Block-based reversible data embedding. Signal Process., 89: 556-569.

Hwang, J.H., J.W. Kim and J.U. Choi, 2006. A Reversible Watermarking Based on Histogram Shifting. In: Digital Watermarking, Shi, Y.Q. and B. Jeon (Eds.) Springer-Verlag, Berlin, Heidelberg, ISBN: 978-3-540-48825-5, pp: 348-361.

Jung, K.H. and K.Y. Yoo, 2009. Data hiding method using image interpolation. Comput. Standards Interfaces, 31: 465-470.

Kim, K.S., M.J. Lee, H.Y. Lee and H.Y. Lee, 2009. Reversible data hiding exploiting spatial correlation between sub-sampled images. Pattern Recognit., 42: 3083-3096.

Lin, C.C., W.L. Tai and C.C. Chang, 2008. Multilevel reversible data hiding based on histogram modification of difference images. Pattern Recogn., 41: 3582-3591.

Liu, L., T.S. Chen, C. Cao, X. Wen and R.S. Xie, 2013. A novel data embedding method using random pixels selecting. Inform. Technol. J., 12: 1299-1308.

Luo, L., Z. Chen, M. Chen, X. Zeng and Z. Xiong, 2010. Reversible image watermarking using interpolation technique. IEEE Trans. Inform. Forensics Security, 5: 187-193.

Luo, H., F.X. Yu, H. Chen, Z.L. Huang, H. Li and P.H. Wang, 2011. Reversible data hiding based on block median preservation. Inform. Sci., 181: 308-328.

Ni, Z., Y.Q. Shi, N. Ansari and W. Su, 2006. Reversible data hiding. IEEE Trans. Circ. Syst. Video Technol., 16: 354-362.

Pevny, T., P. Bas and J. Fridrich, 2010. Steganalysis by subtractive pixel adjacency matrix. IEEE Trans. Inform. Forensics Secur., 5: 215-224.

Provos, N. and P. Honeyman, 2003. Hide and seek: An introduction to steganography. IEEE Secur. Privacy, 1: 32-44.

Sachnev, V., H.J. Kim, J. Nam, S. Suresh and Y.Q. Shi, 2009. Reversible watermarking algorithm using sorting and prediction. IEEE Trans. Circ. Syst. Video Technol., 19: 989-999.

Tian, J., 2003. Reversible data embedding using a difference expansion. IEEE Trans. Circ. Syst. Video Technol., 13: 890-896.

Tsai, P.Y., Y.C. Hu and H.L. Yeh, 2009. Reversible image hiding scheme using predictive coding and histogram shifting. Signal. Process., 89: 1129-1143.

Wang, H. and S. Wang, 2004. Cyber warfare: Steganography vs. steganalysis. Commun. ACM, 47: 76-82.

Yang, C.Y., C.H. Lin and W.C. Hu, 2012. Reversible data hiding for high-quality images based on integer wavelet transform. J. Inform. Hiding Multimedia Signal Process., 3: 142-150.

Zhang, X. and S. Wang, 2006. Efficient steganographic embedding by exploiting modification direction. IEEE Commun. Lett., 10: 781-783.