Use of Three-Form Module Substitution to Hide Data

Ching-Yu Yang
Department of Computer Science and Information Engineering
National Penghu University Penghu, Taiwan, 880, ROC
chingyu@npu.edu.tw

Abstract

A simple, adaptive data hiding method by employing the three-form module substitution is proposed. In accordance with the local variance of the blocks, a variety of secret bits are effectively embedded in each block via one of the three-form module substitution. Experiments show that both the PSNR and hiding rate generated by the proposed method are better than those generated by the reported methods. Since the perceived quality of the resulting mixed images is good, the proposed method has an extra merit of being attracted hardly by the third parties.

1: Introduction

With the explosive growth of computer networks and the popularity of the world wide web (WWW), people are capable of exchanging data efficiently among the parties. In other words, it provides a convenient way for people to transmit their digital multimedia data on the Internet. However, data could be unauthorized tampered, illegal intercepted, or even damaged by the hackers. Therefore, how to prevent the third parties from extracting or attacking data becomes a great concern in data hiding. Recently, several researchers have presented the related papers in the literature [1-5]. On the other hand, some data hiding methods [6-8] with fixed number of secret bits replacement have been proposed to enlarge the hiding space while maintaining a certain degree of the resulting perceptual quality.

In this paper, the author suggests an adaptive data hiding method based on the three-form module substitution. More specifically, the secret bits to be embedded in the block are embedded by one of the three module substitutions, in which the selected module substitution is determined by the base-value of the block. Whereas, the base-value b_k of a block is evaluated by $b_k = \max_{0 \le j \le n \times n-1} q_{kj} - \min_{0 \le j \le n \times n-1} q_{kj} + 1$, where $\{q_{kj}\}_{j=0}^{(n \times n)-1}$ indicates the *k*th $n \times n$ block of a host image and q_{kj} represents for the *j*th pixel value in the *k*th block.

The rest of the paper is organized as follows. The proposed three-form module substitution, namely, Mod *u*, Mod *v*, and Mod *w* substitution are described in Section 2. A fine-labeling policy is also included. In addition, the algorithms of the encoding part and decoding part of the proposed method are presented. Experimental results are demonstrated in Section 3. Finally, a brief conclusion is given in Section4.

2: Proposed method

2.1: Mod *u* substitution

Assume that the Mod *u* substitution is to be applied to hide data. Take $L = \lfloor (n \times n) \log_2 u \rfloor$ bits from the binary data. (Here, the value of L is evalated by the *Property 1* stated later.) Convert this binary numebr to a (base *u*) value of n^2 digits, and each digit is in the range {0, 1, 2, …, *u*-1}. To hide a data digit $d \in$ {0, 1, 2, …, *u*-1} in a pixel with gray value *x*, we first evaluate

$$g_0 = (x - x_{\bmod u}) + d. \qquad (1)$$

Then let

$$g_0^+ = g_0 + u \qquad (2)$$

and

$$g_0^- = g_0 - u. \qquad (3)$$

Choose from the Eqs. (1)-(3) the one whose distortion to *x* is the smallest, and call it *g*. Then, replace the gray value *x* of the host pixel by the new value *g*. Note that the data digit *d* can be extracted later easily because

$$d = (g) \bmod u = (g_0) \bmod u \cdot \qquad (4)$$

Property 1. The number of bits to be placed in the *k*th $n \times n$ block of a host image is $L = \lfloor (n \times n) \log_2 \phi \rfloor$, where ϕ is the value used by a module substitution.

Proof. The largest n^2 -digit number in the base-ϕ number system is $(\phi-1, \phi-1, ..., \phi-1)_\phi$, whose decimal value is $\sum_{j=0}^{n^2-1} (\phi-1)\phi^j = (\phi-1) \cdot \frac{(\phi^{n^2}-1)}{(\phi-1)} = (\phi^{n^2}-1)$. Similarly, the largest *L*-bit binary number is $(1,1,...,1)_2$, whose decimal value is $2^L - 1$. In order that $2^L - 1$ can be stored as a base-ϕ number using *n* digits, we require $2^L - 1 \le \phi^{n^2} - 1$, that is, $2^L \le \phi^{n^2}$. In other words, $L \le n^2 \log_2 \phi$. Now, the maximal integer L satisfying $L \le n^2 \log_2 \phi$ is $L = \lfloor n^2 \log_2 \phi \rfloor$ □

It is clear from *Property 1* that the length of bits to be embedded into the *k*th $n \times n$ block is determined by the module value ϕ.

After the embedding, to help the decoder identify the block type, we must label the resulting mixed block so that the new base value c'_k of the block is satisfying $c'_k < \tau$. Here, τ is a control parameter. To do labeling, we add or subtract full multiples of *u* to or from the gray values of certain pixels. This adding *u* or subtracingt *u* acction will not affect the value of the extracted data, because the decoder will take $gray_{\mathrm{mod}\,u}$ as the extracted value for each pixel of the mixed block, and $g_{\mathrm{mod}\,u} = (g \pm u)_{\mathrm{mod}\,u}$ for any value *g*.

2.2 Fine-labeling policy

To reduce further the distortion caused by the module substitution, a fine-labeling policy is introduced in this subsection. Without loss of generality, let H_{kb} be the *k*th host block of a host image, namely, $H_{kb} = \{g_i \mid i = 1,2,...,|H_{kb}|\}$ with $|H_{kb}| = n^2$ gray values and S_{kb} be the corresponding (temporary) stego-block $S_{kb} = \{s_i \mid i = 1,2,...,|S_{kb}|\}$ with $|S_{kb}| = n^2$ stego-pixels generated by Mod *u* substitution. Also let the (sparse) *k*th minus-block M_{kb} be the $M_{kb} = \{m_i \mid 1 \le i \le n^2\}$ with $m_i = s_i - u$ if $|s_i - u - g_i| = \min\{|s_i + u - g_i|, |s_i - g_i|, |s_i - u - g_i|\}$, otherwise, m_i is a null; and the (sparse) *k*th plus-block P_{kb} be the $P_{kb} = \{p_i \mid 1 \le i \le n^2\}$ with $p_i = s_i + u$ if $|s_i + u - g_i| = \min\{|s_i + u - g_i|, |s_i - g_i|, |s_i - u - g_i|\}$, otherwise, p_i is a null. The details of labeling policy are specified in the following algorithm.

Algorithm. The design of fine-labeling procedure for Mod *u* substitution.

Input: A stego-block S_{kb}, a minus-block M_{kb}, a plus-block P_{kb}, a control parameter *τ*, and an integer *u*.

Output: A stego-block of which base-value c'_k is satisfying $c'_k < \tau$.

Step 1. Choose the minimum value $m_{\min}$ from M_{kb}, presumably, $m_{\min}$ was located at the *r*th pixel of M_{kb} with $1 \le r \le n^2$, add *u* to $m_{\min}$ to obtain $\tilde{m}_{\min} = m_{\min} + u$, and compute the base-value c'_i of S_{kb} with $\tilde{m}_{\min}$ but excluding the stego-pixel s_r.

Step 2. Choose the maximum value $m_{\max}$ from P_{kb}, presumably, $m_{\max}$ was located at the *q*th pixel of P_{kb} with $1 \le q \le n^2$, subtract *u* from $m_{\max}$ to obtain $\tilde{m}_{\max} = m_{\max} - u$, and compute the base-value c'_j of S_{kb} with $\tilde{m}_{\max}$ but excluding the stego-pixel s_q.

Step 3. If both c'_i and c'_j are satisfying $c'_i < \tau$ and $c'_j < \tau$, respectively, then we employ the one from Steps 1 and 2 with the minimum distortion, and replace s_r (or s_q) by $\tilde{m}_{\min}$ (or $\tilde{m}_{\max}$) in accordance with which step was taken and go to Step 9. However, if either $c'_i < \tau$ or $c'_j < \tau$ is true, then the corresponding step was chosen. Subsequently, we either replace the stego-pixel s_r by $\tilde{m}_{\min}$ when $c'_i < \tau$ or replace the stego-pixel s_q by $\tilde{m}_{\max}$ when $c'_j < \tau$ and go to Step 9.

Step 4. Evaluate the base-value c'_k by $c'_k = \tilde{m}_{\max} - \tilde{m}_{\min} + 1$. If $c'_k < \tau$, then substitute $\tilde{m}_{\min}$ and $\tilde{m}_{\max}$ for s_r and s_q, respectively, and go to Step 9.

Step 5. Choose the minimum value $s_{\min}$ of S_{kb}, add *u* to $s_{\min}$ to get $\tilde{s}_{\min} = s_{\min} + u$, and compute the base-value c'_l of S_{kb} with $\tilde{s}_{\min}$ instead of $s_{\min}$.

Step 6. Choose the maximum value $s_{\max}$ of S_{kb}, subtract *u* from $s_{\max}$ to get $\tilde{s}_{\max} = s_{\max} - u$, and compute the base-value c'_m of S_b with $\tilde{s}_{\max}$ instead of $s_{\max}$.

Step 7. If both c'_l and c'_m are satisfying $c'_l < \tau$ and $c'_m < \tau$, respectively, then we take the one from Steps 5 and 6 with the minimum distortion, and replace $s_{\min}$ (or $s_{\max}$) of S_{kb} by $\tilde{s}_{\min}$ (or $\tilde{s}_{\max}$) in accordance with which step was taken and go to Step 9. However, if either $c'_l < \tau$ or $c'_m < \tau$ is true, then the corresponding step was chosen and $s_{\min}$ of S_{kb} was replaced by $\tilde{s}_{\min}$ when $c'_l < \tau$ or $s_{\max}$ of S_{kb} was replaced by $\tilde{s}_{\max}$ when $c'_m < \tau$ and go to Step 9.

Step 8. Substitute $\tilde{s}_{\min}$ and $\tilde{s}_{\max}$ for $s_{\min}$ and $s_{\max}$ of S_{kb}, respectively, and compute the new base-value c'_k of S_{kb}. If $c'_k < \tau$, then go to Step 9, otherwise, repeat to Step 5.

Step 9. Stop.

2.3: Mod *v* and Mod *w* substitutions

The main goal of employing Mod *v* and Mod *w* substitutions into the proposed method is to promote further the hiding rate. That is, the base-value b_k of a block being greater than or equal to τ is either encoded by Mod *v* substitution or Mod *w* substitution. However, to help the decorder distinguish which substitution was employed, a pixel (called flag-pixel) at the top-left of the block was used to as a flag to identify the block type.

Before the embedding, the least significant bit (LSB) of the flag-pixel of the block is first set to be either zero if the base-value b_k of a block being satisfied $\tau \le b_k < 2\tau$ or one if $b_k \ge 2\tau$. In other words, if the LSB of the flag-pixel is zero then the block is embedded by Mod *v* substitution. Otherwise, the block is embedded by Mod *w* substitution. The secret bits are then embedded in the remaining $n^2 - 1$ pixels of the block. Finally, as described in Section 2.1, to ensure the decoder for extracting data correctly, a similar labeling procedure has to be imposed on the mixed block. However, the base-value c_k of the mixed block must be labeled to a new base c'_k so as to satisfying $c'_k \ge \tau$ when c_k belows τ.

2.4: Embedment and extraction procedures

The encoding part of the proposed method is first summarized in the following algorithm.

```
Algorithm. The design of hiding method by using the three-form module substitution.
Input: A test secret image treated as a (very long but
    finite) binary sequence, a control parameter $\tau$, and
    three integers u, v, and w with $u \le v \le w$ for module substitution.
Output: A mixed image contains the secret data.
Methods:
    Step 1: Input a block not process yet, and compute
        the base-value $b_k$ of the block.
    Step 2: If $b_k < \tau$, then embed $\lfloor n^2 \log_2 u \rfloor$ bits of
        data to the block by Mod u substitution, label
        this mixed block so that the new base-value
        $c'_k$ is satisfying $c'_k < \tau$, and go to Step 1.
    Step 3: If $b_k \ge \tau$, then the LSB $\alpha$ of the
        flag-pixel in the block is set to be either 0 if
        $\tau \le b_k < 2\tau$ or 1 if $b_k \ge 2\tau$. Subsequently,
        data bits are encoded in accordance with the
        following two substeps.
      Step 3a: If $\alpha = 0$, then embed $\lfloor (n^2 - 1)\log_2 v \rfloor$
        bits of data to the block by Mod v
        substitution, label this mixed block so that
        the new base-value $c'_k$ is satisfying
        $\tau \le c'_k < 2\tau$, and go to Step 1.
      Step 3b: If $\alpha = 1$, then embed $\lfloor (n^2 - 1)\log_2 w \rfloor$
        bits of data to the block by Mod w
        substitution, label this mixed block so that
        the new base-value $c'_k$ is satisfying
        $c'_k \ge 2\tau$, and go to Step 1.
    Step 4: Stop.
```

The decoding part of the proposed method is much simpler than that of the encoding part. First, read in the next $n \times n$ block of the mixed image that is not processed yet, and compute the base-value c_k of the block. If $c_k < \tau$, then the n^2-digit are extracted from the block by Mod *u* and being converted to $\lfloor n^2 \log_2 u \rfloor$ bits. Otherwise, the LSB α of the flag-pixel in the block was picked up. The n^2-1 digits are extracted from the block either by Mod *v* if α is equal to zero or by Mod *w* if α is equal to one. Afterward, the extracted digits are converted to the corresponding $\lfloor (n^2 - 1)\log_2 v \rfloor$ and $\lfloor (n^2 - 1)\log_2 w \rfloor$ bits, respectively. The above procedure is repeated until the embedded bits are completely extracted.

3: Experimental results

Three 512 × 512 gray scale images, namely, *Lena*, *Peppers*, and *Baboon* were used as host images. During the simulations, one of three host images *Lena* was also used as a test image. The block size is 3 × 3. One of the three integer *u* used her is 3, whereas the values of the other two integers *v*, *w* with *w=1.5 v*, and a control parameter τ are not (necessary) fixed. The PSNR generated by the proposed method using various *v* and *w* in the range $4 \le \tau \le 10$ was illustrated in Fig. 1. The quantitative measure PSNR is defined by

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE}, \qquad (5)$$

where $MSE = \frac{1}{MN} \sum_{i=1}^{N} \sum_{j=1}^{M} (\hat{x}(i,j) - x(i,j))^2$ if the image size is $M \times N$. Here $x(i,j)$ and $\hat{x}(i,j)$ denote the pixel values of the original image and the mixed image, respectively. It can be seen from Fig. 1 that the PSNR is linearly increased as the value of τ being enlarged. In addition, the hiding rate generated by the proposed method was shown in Fig. 2. From Fig. 2, we can see that the hiding rate is gradually decreased as the value of τ being increased. From Figs. 1 and 2, we can conclude that the larger the τ, the larger portion of the blocks are encoded by Mod *u* substitution which means that a lower hiding rate (or a higher PSNR) is obtained, and vice versa. Moreover, if we let the value of *v* be fixed and *u* be a variable, it can be also found that a similar result is acquired by the proposed method. Fig. 3 illustrates the tradeoff between the PSNR and hiding rate that generated by the proposed method. Finally, to increase further the hiding rate, the second- and third-rightmost bits in a flag-pixel of the block could be used to hide two more bits. Simulations show that the hiding rate for the proposed method can be achieved to a scale of 2.3% bettern than that for the original one while the PSNR is dropped around 0.50 dB.

A mixed image generated via the proposed method by hiding part of the test image into the host image *Lena* was depicted in Fig. 4(a). The value of τ is 8, while u = 3, v = 8, and w = 12. It can be seen from Fig. 4(a) that the perceived quality is good at a hiding rate of 0.3167 (with PSNR = 40.33). The absolute-value-difference image generated from Fig. 4(a) was also given in Fig. 4(b). The absolute-value-difference at each pixel has been amplified in order to view clearly. For example, the "brighter" pixel in Fig. 4(b) indicates bigger error there, whereas the darkest pixel in Fig. 4(b) indicates no error there.

Several adaptive data hiding schemes [10-12] are also used to compare with the proposed method. The PSNR and hiding rate provided by those methods are tabulated in Table 1. The hiding rate is a percentage defined as a ratio between the number bits being hidden and 8 × 512 × 512. It can be seen from Table 1 that the proposed method has the best PSNR value at various hiding rate among these methods.

4: Conclusions

In this paper, the author uses three forms of module substitution to obtain a simple, adaptive data hiding method. In accordance with the base-value of a block, various data bits are effectively embedded in the block via the three-form module substitution. A fine-labeling policy is also employed to alleviate the distortion during data embedment. Simulations show that both the PSNR and hiding rate for the proposed method are better than those for the reported schemes. Since the perceptual quality of the mixed images is good, the third parties are hardly aware of the existence of the embedded message. Notice that the proposed method is not designed for resisting attacking, any of image processing operations imposed on the mixed images would block a successful extraction of the data at the receiver site. However, another important usage of the proposed method can be found in image authentication.

References

[1] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process.* 6, 1997, pp. 1673-1686.

[2] I. J. Cox and J. P. Linnartz, "Some general methods for tampering with watermarks," *IEEE J. Selected Area Comm.* 16, 1998, pp. 587-593.

[3] S. Katzenbeisser and F.A.P. Petitcolas (Eds.), *Information Hiding Techniques for Steganography and Digital Watermarking*, MA: Artech House, 2000.

[4] C. T. Li, "Digital fragile watermarking scheme for authentication of JPEG images," *IEE Proc. Vision Image Signal Process*. 151, 2004, pp. 460-466.

[5] J.S. Pan, H.C. Huang and L.C. Jain (Eds.), *Intelligent Watermarking Techniques*, Singapore: World Scientific, 2004.

[6] R. Z. Wang, C. F. Lin, and J. C. Lin, "Image hiding by LSB substitution and genetic algorithm," *Pattern Recognition* 34, 2001, pp. 671-683.

[7] C. C. Chang, J. Y. Hsiao, and C. S. Chen, "Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy," *Pattern Recognition* 36, 2003, pp. 1583-1595.

[8] Y. C. Hu and M. H. Lin, "Secure image hiding scheme based upon vector quantization," *Int. J. Pattern Recognition and Artificial Intell.* 18, 2004, pp. 1111-1130.

[9] C. K. Chan, and L. M. Chen, "Hiding data in images by simple LSB substitution," *Pattern Recognition* 37, 2004, pp. 469-474.

[10] S. H. Liu, T. H. Chen, H. X. Yao, and W. Gao, "A variable depth LSB data hiding technique in images," *Proc. 3rd Int. Conf. Machine Learning and Cybernetics,* Shanghai, August 2004, pp. 26-29.

[11] X. Zhang and S. Wang, "Steganography using multiple-base notational system and human vision sensitivity," *IEEE Signal Processing Letters* 12, 2005, pp. 67-70.

[12] H. C. Wu, N. I. Wu, C. S. Tsai, and M. S. Hwang, "Image steganographic scheme based on pixel-value differencing and LSB replacement methods," *IEE Proc. Vision Image Signal Process*. 152, 2005, pp. 611-615.

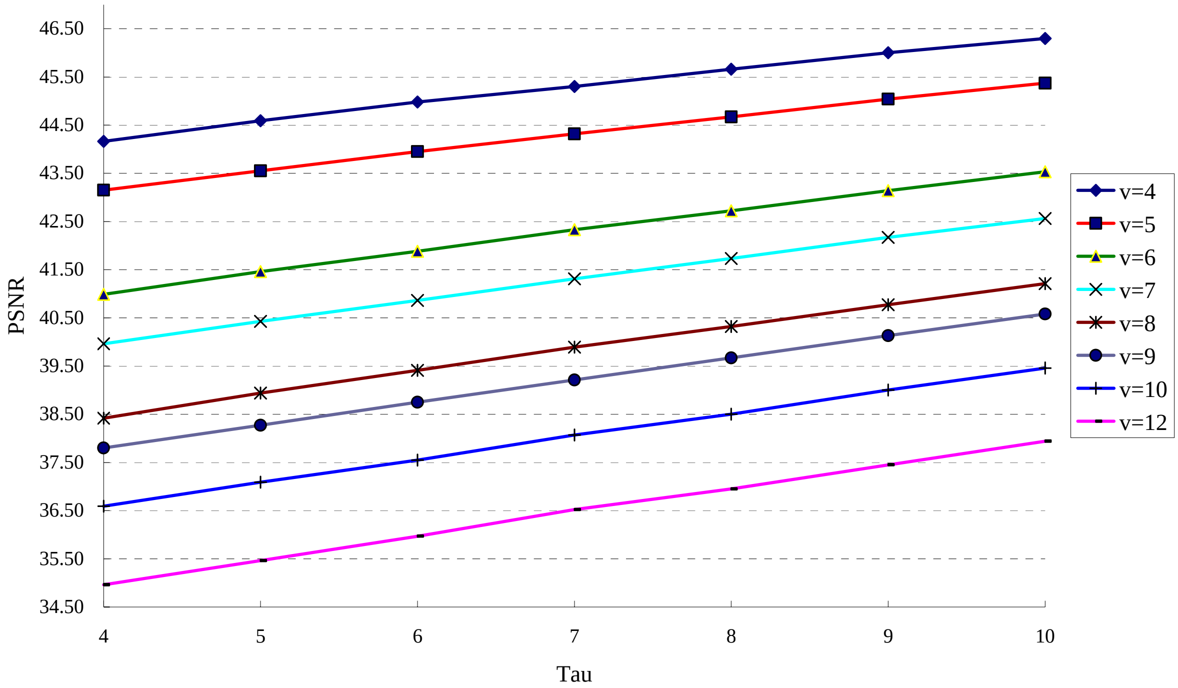


Fig. 1. The PSNR generated by the proposed method via various *v*, *w* with *w*=1.5 *v* in the range $4 \le \tau \le 10$.

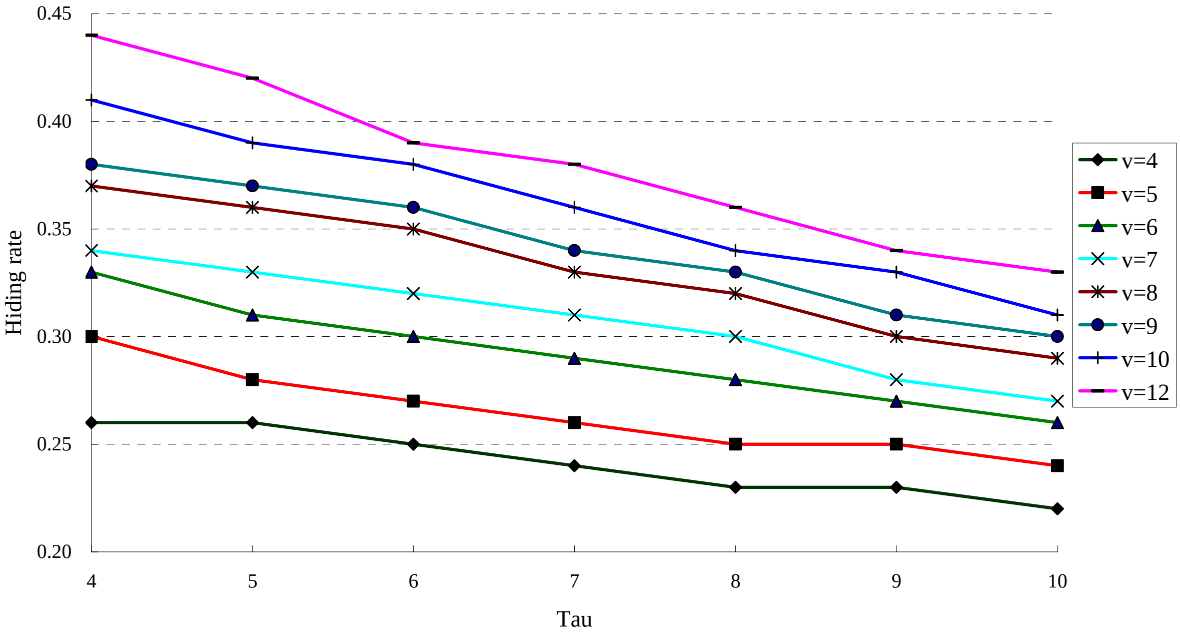


Fig. 2. The hiding rate generated by the proposed method via various *v*, *w* with *w*=1.5 *v* in the range $4 \le \tau \le 10$.

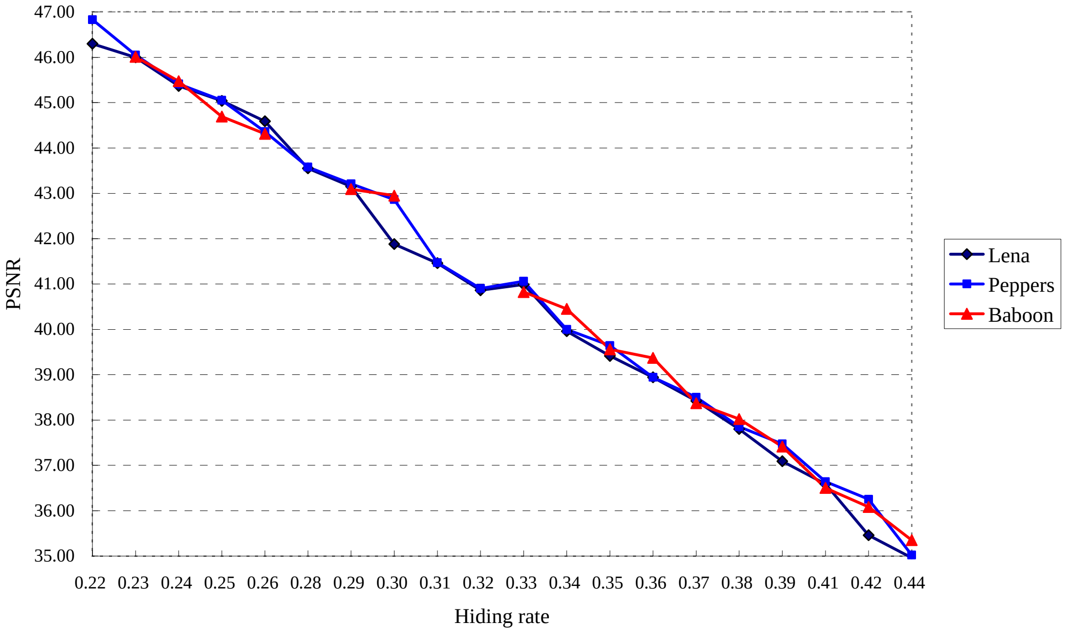
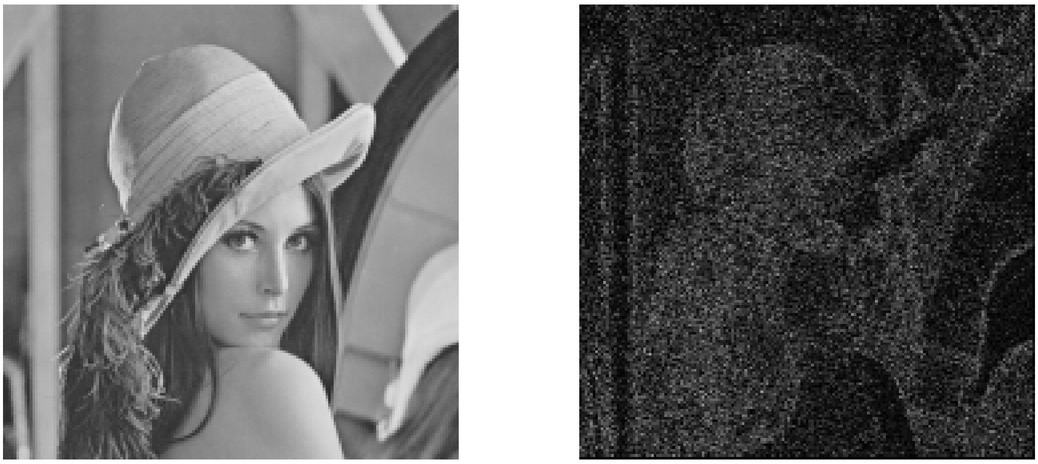


Fig. 3. The relationship between PSNR and hiding rate for the proposed method.



(a) **(b)**

Fig. 4. The resulting images genertaed by the proposed method. (a) The mixed image is obtained using $\tau = 8$, $u = 3$, $v = 8$, and $w = 12$ while the PSNR is 40.33 and the hiding rate is 0.3167. (b) The absolute-value-difference image generated from Fig. 4(a).

Table 1. PSNR generated by the proposed method and other adaptive LSB techniques on the image *Lena* at various hiding rate.

Methods	Hiding rate					
	0.20	0.25	0.30	0.35	0.37	0.44
Liu *et al.* [10]	45.12	-	-	-	-	32.66
Zhang and Wang [11]	43.50	41.20	39.90	38.10	-	-
Wu *et al.* [12]	-	38.80			36.16	-
Proposed method	48.40	45.04	41.88	39.41	38.42	34.96