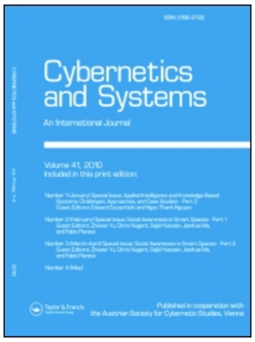

Taylor & Francis
Taylor & Francis Group

Cybernetics and Systems

An International Journal

ISSN: 0196-9722 (Print) 1087-6553 (Online) Journal homepage: http://www.tandfonline.com/loi/ucbs20

Adaptive Data Hiding, Using Pixel-Value-Differencing and LSB Substitution

Masoumeh Khodaei, Bahram Sadeghi Bigham & Karim Faez

To cite this article: Masoumeh Khodaei, Bahram Sadeghi Bigham & Karim Faez (2016) Adaptive Data Hiding, Using Pixel-Value-Differencing and LSB Substitution, Cybernetics and Systems, 47:8, 617-628, DOI: 10.1080/01969722.2016.1214459

To link to this article: https://doi.org/10.1080/01969722.2016.1214459

Published online: 26 Aug 2016.

CYBERNETICS AND SYSTEMS: AN INTERNATIONAL JOURNAL
2016, VOL. 47, NO. 8, 617–628
http://dx.doi.org/10.1080/01969722.2016.1214459

Taylor & Francis
Taylor & Francis Group

Adaptive Data Hiding, Using Pixel-Value-Differencing and LSB Substitution

Masoumeh Khodaei[a], Bahram Sadeghi Bigham[a], and Karim Faez[b]

[a]Department of Computer Science and Information Technology, Institute for Advanced Studies in Basic Sciences (IASBS), Zanjan, Iran; [b]Electrical Engineering Department, Amirkabir University of Technology, Tehran, Iran

ABSTRACT
With the development of Internet communications, the security of message sending on the Internet has become very important. This article proposes a new adaptive data hiding method with a large data-embedding capacity for gray-scale images to raise the security of sending a message between sender and receiver in networks. At first, the image is divided into some blocks consisting of two consecutive pixels. If the values of both pixels are small, fewer secret bits will be embedded within the two pixels, otherwise, the difference value of two pixels is calculated, and according to the obtained difference value, the method will estimate the number of embedding bits into LSBs of two pixels. This number is adaptive and depends on the range to which the difference value belongs. A readjusting phase is presented to keep the difference of value pixels in the same range before and after embedding. Experimental results show that our method has increased the capacity of embedding bits in comparison with the several other methods.

KEYWORDS
Adaptive data hiding; LSB substitution; pixel-value-differencing; steganography

Introduction

Data hiding is a useful technique to embed secret data into cover media for the purposes of copyright protection, authentication, fingerprinting, secret communication, etc. Data hiding techniques are applied in two fields, watermarking and steganography. Watermarking is a reversible data hiding technique to embed secret data called a *mark* into host media, which is referred to as lossless data hiding, such that the host media can be completely recovered after extracting the secret data. Steganography is a technique to conceal secret data into host media so that the existence of a secret message into host media is imperceptible and invisible to the human visual systems. In this study, the cover media that is utilized for data

CONTACT Bahram Sadeghi Bigham b_sadeghi_b@iasbs.ac.ir Department of Computer Science and Information Technology, Institute for Advanced Studies in Basic Sciences (IASBS), No. 444, Prof. Yousef Sobouti Blvd, PO Box 45195-1159, Zanjan 45137-66731, Iran.
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/ucbs.

embedding is an image named the *cover image* and the image carrying the secret data is called the *stego-image*.

Many approaches for watermarking are presented in the literature. Several approaches are based on difference expansion, which was introduced by Tian (2003). Some of the methods utilize the image histogram for data embedding, providing a large embedding capacity and a high quality of marked-images (Chang and Tai 2012). Also, other methods are presented that require no location map to record the marked pixels' positions.

In the steganography field, one of well-known steganographic methods is Least-Significant-Bit (LSB) substitution, which utilizes the LSBs of cover-image pixels to embed secret data bits (Wang, Lin, and Lin 2001; Chan and Cheng 2003; Khodaei and Faez 2010). This method has high embedding capacity, but the security of the method against steganalysis attacks such as an RS attack is low. To improve the security of data embedding, Wu and Tsai presented a new concept based on human vision properties (Wu and Tsai 2003). According to human vision properties, the tolerance of changes in edge areas of the host image is higher than in smooth areas. Therefore, more data can be embedded in the edge areas than in smooth areas. This method is known as the pixel-value-differencing (PVD) method; it utilizes the difference value between two consecutive pixels to determine the number of secret bits that can be embedded into cover-image pixels (Yang et al. 2011). This concept is applied in several studies to get the stego-image with high quality and larger capacity embedding (Wu et al. 2005; Chang et al. 2008; Yang, Weng, and Wang 2008).

In this article, we present a new adaptive steganographic method, using PVD to hide secret data in a gray-scale cover image. This method has two phases. In the first phase, secret data are hidden in LSB bits of cover-image pixels. The number of secret embedding bits is variable and is related to the value of the cover-image pixels and the difference value of two consecutive pixels in the cover image. We apply the range table to obtain the hiding capacity of each pixel. Also, we introduce a readjusting process to adopt the difference value of pair-successive pixels and the remaining difference value in the same range, before and after embedding secret data. The method has high hiding capacity and acceptable image quality. In the second phase, we extract the secret data bits from LSBs of the stego-image pixels, exactly.

The remaining parts of the article are organized as follows. In "Related Work," the proposed method will be presented. In "Proposed Method" the experimental results of the proposed method will be illustrated and compared with the modulus function method that is proposed by Lee and Chen (2010) and the adaptive LSB data hiding method that is presented by Yang et al. (2008). Conclusions shall be given in the final section.

Related Work

The PVD and LSB Replacement Method

Wu et al. presented a useful method using PVD and the LSB replacement technique for embedding secret data in gray-level images in 2005 (Wu et al. 2005). In this method, the cover image is partitioned into some nonoverlapping blocks that have two consecutive pixels, denoted by p_i, p_{i+1}. Moreover, a range table is introduced to divide the range [0, 255] into two levels as "lower level" and "higher level." An example for the range table with Div = 15 is shown in Figure 1. The Div denotes the location of ranges divisions. According to Figure 1, the lower level consists of $R_1 = [0, 15]$, and the higher level includes $R_2 = [16, 31]$, $R_3 = [32, 63]$, $R_4 = [64, 127]$, and $R_5 = [128, 255]$. The lower bound value of R_k is called l_k, and the upper bound values of R_k are called u_k, (i.e.,. $R_k \in [l_k, u_k]$, $k = 1,2,\ldots 5$). Therefore, the width of R_k is denoted as $|R_k|$. The difference value d_i for each block is given by $d_i = |p_i - p_{i+1}|$. Then, two cases may take place:

Case 1. If the difference value d_i is assigned to the lower-level, the LSB substitution method is applied to embed 6-bit secret data into the two pixels as follows:

Suppose that the 6-bit secret data is $S = a_1, a_2, a_3, a_4, a_5, a_6$.

1. Replace a_1, a_2, a_3 into 3-LSB of p_i to get p'_i.
2. Replace a_4, a_5, a_6 into 3-LSB of p_i to get p'_{i+1}.
3. Compute the new difference value d'_i as $d'_i = |p'_i - p'_{i+1}|$.
4. If the difference value d'_i belongs to the higher level, complete the readjusting operation as

$$(p'_i, p'_{i+1}) = \begin{cases} (p'_i - 8,\ p'_{i+1} + 8), & \text{if } p'_i \geq p'_{i+1} \\ (p'_i + 8,\ p'_{i+1} - 8), & \text{if } p'_i < p'_{i+1} \end{cases} . \tag{1}$$

Case 2. If the difference value d_i is assigned to the higher level, the pixel-value differencing (PVD) method (Wu et al. 2005) is used to conceal the secret bits in the two consecutive pixels as follows:

1. Refer to the range table and find out the range to which d_i belongs (e.g., $R_k \in [l_k, u_k]$).
2. Compute how many secret bits t_k can be embedded in the pair pixel p_i and p_{i+1} using $t_k = \lfloor \mathrm{Log}_2(|R_k|) \rfloor$
3. Read t_k bits from the binary secret data and convert the binary bit-stream into a decimal value s_i. For example, if bit-stream = 011, then $s_i = 3$.

Lower-level ←---→	←------------------ Higher-level ------------------→			
$R_1 = [0, 15]$	$R_2 = [16, 31]$	$R_3 = [32, 63]$	$R_4 = [64, 127]$	$R_5 = [128, 255]$

Figure 1. Example of division of lower level and higher level in the method of Wu et al. (2005) (Div = 15).

4. Calculate the new difference value d'_i as $d'_i = l_i + s_i$.
5. Compute $m = | d_i - d'_i |$ and adjust the values of p_i and p_{i+1} by

$$(p'_i, p'_{i+1}) = \begin{cases} (p_i + \lceil m/2 \rceil, p_{i+1} - \lfloor m/2 \rfloor), & \text{if } p_i \geq p_{i+1} \text{ and } d'_i > d_i, \\ (p_i - \lfloor m/2 \rfloor, p_{i+1} + \lceil m/2 \rceil), & \text{if } p_i < p_{i+1} \text{ and } d'_i > d_i, \\ (p_i - \lceil m/2 \rceil, p_{i+1} + \lfloor m/2 \rfloor), & \text{if } p_i \geq p_{i+1} \text{ and } d'_i \leq d_i, \\ (p_i + \lceil m/2 \rceil, p_{i+1} - \lfloor m/2 \rfloor), & \text{if } p_i < p_{i+1} \text{ and } d'_i \leq d_i, \end{cases} \quad (2)$$

Assume that Div = 15, $p_i = 54$, $p_{i+1} = 68$, and secret data $s = (010111)_2$. The difference value d_i is calculated as $|54 - 68| = 14$, that are assigned to lower-level. As a result, the 3-bit LSB substitution approach is applied to embed the secret bits $s = (010111)_2$ into p_i and p_{i+1}. After embedding, we obtain $p'_i = 50$ and $p'_{i+1} = 71$. Therefore, $d'_i = 21$, which belongs to the higher level. So, the readjusting operation is performed by Eq. (1), and $p'_i = 58$, $p'_{i+1} = 63$ is obtained. Then, the new difference value is as $d'_i = |58 - 63| = 5$, which belongs to the lower level yet.

The Adaptive Data Hiding Method

Yang et al. presented an adaptive data hiding method based on pixel-value differencing for gray-scale images (Yang, Weng, and Wang 2008). This method increased the embedding capacity while maintaining the high quality of stego-image. Moreover, this method utilized the concept in which the edge areas can tolerate more changes than the smooth areas can. At first, the cover image is partitioned into several nonoverlapping blocks of two successive pixels, say p_i, p_{i+1}. Two dividing cases are shown in Figure 2. According to Figure 2(a), in the first case (*l-h* division), the range [0, 255] is divided into "lower level" and "higher level" (div = 15) for which the lower level consists of $R_1 = [0, 15]$ and the higher level includes $R_2 = [16, 255]$. Also, in the second case (*l-m-h* division), as shown in Figure 2(b), the range [0,255] is divided into "lower level," "middle level," and "higher level" (div1 = 15 and div2 = 31). The lower level consists of $R_1 = [0, 15]$, the middle level contains $R_2 = [16, 31]$, and the higher level includes $R_3 = [32, 255]$.

Then, the difference value d_i for each block is computed by $d_i = |p_i - p_{i+1}|$ and the range to which d_i belongs is located. After that, the number of secret

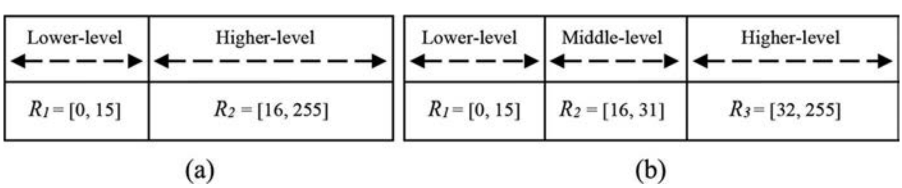

Figure 2. Two dividing cases of the method of Yang et al. (2008): (a) l-h division with "lower level" and "higher level," (b) l-m-h division with lower level, middle level, and higher level.

bits that can be embedded into LSBs of p_i and p_{i+1} is determined. For instance, in *l-m-h* division, if d_i belongs to the lower level, middle level, and higher level, *l-bits, m*-bits, and *h*-bits LSB substitution will be applied to embed the secret bits into pixels p_i and p_{i+1}. In *l-h* division, the conditions $l \le \log_2|R_1|$ and $h \le \log_2|R_2|$ are used and the conditions $l \le \log_2|R_1|$, $m \le \log_2|R_2|$ and $h \le \log_2|R_3|$ are utilized for *l-m-h* division. Finally, the new difference value d_i' is calculated by $d_i' = |p_i' - p_{i+1}'|$. If d_i and d_i' fall into different levels, the readjusting operation is performed as follows:

Case 1. $d_i \in$ lower level, $k = l$, and $d_i' \notin$ lower level. If $p_i' \ge p_{i+1}'$, select the better value between $(p_i', p_{i+1}' + 2^k)$ and $(p_i' - 2^k, p_{i+1}')$. Otherwise, if $p_i' < p_{i+1}'$, select the better value between $(p_i', p_{i+1}' - 2^k)$ and $(p_i' + 2^k, p_{i+1}')$.

Case 2. $d_i \in$ middle level, $k = m$, and $d_i' \in'$ lower level. If $p_i \ge p_{i+1}'$, select the better value between $(p_i' + 2^k, p_{i+1}')$ and $(p_i', p_{i+1}' - 2^k)$. Otherwise, if $p_i' < p_{i+1}'$, select the better value between $(p_i', p_{i+1}' + 2^k)$ and $(p_i' - 2^k, p_{i+1}')$.

Case 3. $d_i \in$ middle level, $k = m$, and $d_i' \in$ higher level. If $p_i \ge p_{i+1}'$, select the better value between $(p_i', p_{i+1}' + 2^k)$ and $(p_i' - 2^k, p_{i+1}')$. Otherwise, if $p_i' < p_{i+1}'$, select the better value between $(p_i', p_{i+1}' - 2^k)$ and $(p_i' + 2^k, p_{i+1}')$.

Case 4. $d_i \in$ higher level, $k = h$, and $d_i' \notin$ higher level. If $p_i \ge p_{i+1}'$, select the better value between $(p_i', p_{i+1}' - 2^k)$ and $(p_i' + 2^k, p_{i+1}')$. Otherwise, if $p_i' < p_{i+1}'$, select the better value between $(p_i', p_{i+1}' + 2^k)$ and $(p_i' - 2^k, p_{i+1}')$.

For choosing the best values of p_i' and p_{i+1}', note that (p_i'', p_{i+1}'') can be the best choice when it satisfies three conditions that $d_i'' = |\, p_i'' - p_{i+1}''|$ belongs to the same level of d_i and also, the value of $|\, p_i'' - p_i' \,| + |p_{i+1}'' - p_{i+1}'|$ is smaller and $p_i'', p_{i+1}'' \in [0, 255]$.

For example, assume that we use the 3–4–5 division in Figure 2(a), for which $p_i = 88$, $p_{i+1} = 60$, and the bit-stream of secret data is $s = (11100011)_2$. The difference value d_i is computed as $|88 - 60| = 28$, which belongs to the middle level. Thus, 4-bit LSB substitution is utilized to embed the substream $s_1 = (1110)_2$ into $p_i = 88 = (1011000)_2$ to obtain $p_i' = (1011110)_2 = 94$ and $s_2 = (0011)_2$ into $p_{i+1} = 60 = (111100)_2$ to get $p_{i+1}' = (110011)_2 = 51$. Therefore, $d_i' = |94 - 51| = 43$, which falls into the higher level. Now, the readjusting operation is necessary to adjust the new difference value d_i'. Case 3 has happened. Because we have $p_i' \ge p_{i+1}'$, we should select the better choice between $(94, 51 + 2^4)$ and $(94 - 2^4, 51)$. The better choice is the first one because of $|94 - 88| + |67 - 60| < |78 - 88| + |51 - 60|$. Consequently, $p_i' = 94$, $p_{i+1}' = 67$, and the new difference value $d_i' = |94 - 67| = 27$, which falls into the middle level.

The Proposed Method

In this section, we present our proposed method to embed secret data into the cover images. Our method is based on the concept that sharp edge areas can tolerate larger changes than can smooth areas. Similar to the PVD method

Table 1. The proposed range table.

Ranges	$R1 = [0,15]$	$R2 = [16,63]$	$R3 = [64,127]$	$R4 = [128,255]$
Number of embedding bits	8	10	12	14

(Wu and Tsai 2003), we use the difference value of pixels to determine how many secret bits can be embedded into LSBs of two consecutive pixels.

To estimate the number of secret embedding bits, we design a range table R_i with continuous ranges from 0 to 255 that is introduced in Table 1. We have $R_i \in [l_i, u_i]$ where l_i is the lower bound of R_i, and u_i is the upper bound of R_i. The range table R_i has four ranges $R_1 = [0, 15]$, $R_2 = [16, 63]$, $R_3 = [64,127]$, and $R_4 = [128, 255]$. The proposed method consists of two main phases, the embedding and extracting phases, which are described in the subsections below.

The Embedding Phase

The proposed method uses the image with 256 gray-scales as a cover image. To hide secret data into cover-image pixels, divide the cover image into several nonoverlapping blocks with two consecutive pixels in raster scan manner, say p_i and p_{i+1}. Then, embed n bits of the secret data into p_i and p_{i+1} of each block to the following two cases.

Case 1. If the values of both pixels p_i and p_{i+1} are less than 192 (i.e., $p_i <$ 192 and $p_{i+1} < 192$), read $n = 6$ bits of secret data and embed $k = 3$ first bits of it into k LSBs of pixels p_i to obtain p'_i and $k = 3$ next bits of it into k LSBs of pixels p_{i+1} to obtain p'_{i+1}.

Case 2. If one of the pixel values p_i and p_{i+1} or both are larger or equal to 192 (i.e., $p_i \geq 192$ or $p_{i+1} \geq 192$), hide the secret data into two successive pixels of the cover image according to the following procedure:

Step 1. Calculate the difference value d_i between two consecutive pixels in the block by

$$d_i = |p_i - p_{i+1}|. \tag{3}$$

Step 2. Refer to the range table R_i and find out the range to which d_i belongs. Now, obtain the number of embedding bits n and read n bits of secret data and embed $k = n/2$ first bits of the secret data into k LSBs of pixel p_i to obtain p'_i and $k = n/2$ next bits of it into k LSBs of pixel p_{i+1} to get p'_{i+1}.

Step 3. Calculate the new difference value d'_i, which is given by

$$d'_i = \left|p'_i - p'_{i+1}\right|. \tag{4}$$

Step 4. If d_i and d'_i belong to different ranges, readjust p'_i and p'_{i+1}. Carry out the readjusting process as follows.

1. Compute the modified values of p'_i *and* p'_{i+1} *by*

$$\begin{aligned} p''_i &= p'_i + 2k, \\ p'''_i &= p'_i - 2k, \\ p''_{i+1} &= p'_{i+1} + 2k, \\ p'''_{i+1} &= p'_{i+1} - 2k. \end{aligned} \tag{5}$$

2. Select the optimal value of p'_i and p'_{i+1} by

$$(p'_i, p'_{i+1}) = \text{optimal}\begin{cases} (p'_i, p''_{i+1}) \\ (p'_i, p'''_{i+1}) \\ (p''_i, p'_{i+1}) \\ (p''_i, p''_{i+1}) \\ (p''_i, p'''_{i+1}) \\ (p'''_i, p'_{i+1}) \\ (p'''_i, p''_{i+1}) \\ (p'''_i, p'''_{i+1}) \end{cases}. \tag{6}$$

To choose the optimal values of p'_i and p'_{i+1}, it is essential to notice that the optimal selected values are the values that have minimum difference with original values p_i and p_{i+1}. For example, if $|p_i - p'_i|$ and $|p_{i+1} - p''_{i+1}|$ are minimum in comparison with the other seven values, we choose (p'_i, p''_{i+1}) as optimal value and replace it instead of (p'_i, p'_{i+1}). Also, one of these values or both of them are greater than or equal to 192. Moreover, d_i and d'_i belong to the same range and also, $p'_i \le 255$ and $p'_{i+1} \le 255$.

After embedding every bit of secret data into cover-image pixels, we will obtain a stego-image. For example, suppose that $p_i = 208$, $p_{i+1} = 226$ and a bit-stream of secret data is $s = (1110100011)_2$, which is shown in Figure 3. According to Figure 3, $p_i > 192$ and $p_{i+1} > 192$, Therefore, the difference value d_i is computed as $|208 - 226| = 18$, which belongs to $R_2 = [16,\ 63]$ and $k = 5$. Thus, the proposed method is utilized to embed the substream $s_1 = (11011)_2$ into $p_i = 208 = (11010000)_2$ to get $p'_i = (11011101)_2 = 221$ and $s_2 = (00011)_2$ into $p_{i+1} = 226 = (11100010)_2$ to obtain $p'_{i+1} = (11100011)_2 = 227$. Therefore, $d'_i = |221 - 227| = 6$, which belongs to $R_1 = [0,\ 15]$. Now, the readjusting operation is required to adjust the new difference value d'_i. Then, the modified values of p'_i *and* p'_{i+1} are computed as $p''_i = 221 + 32 = 253$, $p'''_i = 221 - 32 = 189$, $p''_{i+1} = 227 + 32 = 259$, $p'''_{i+1} = 227 - 32 = 195$. The process of the readjusting phase is shown in Figure 3. Finally, $p'_i = 189$, $p'_{i+1} = 227$, and the new difference value $d'_i = |189 - 227| = 38$, which belongs to $R_2 = [16,\ 63]$.

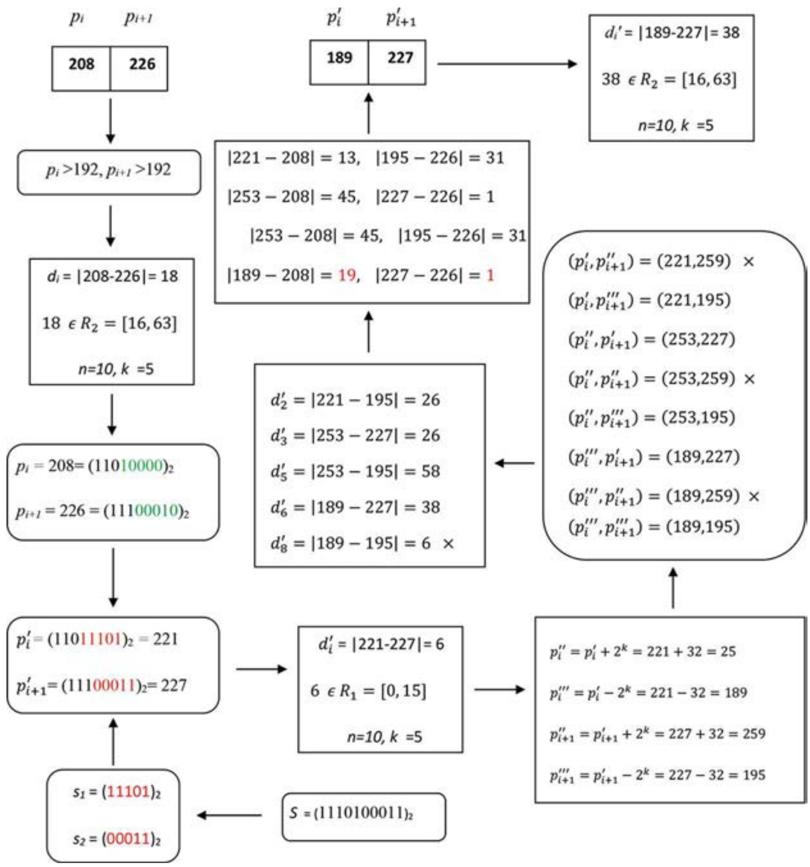

Figure 3. An example for the proposed method.

The Extracting Phase

To extract secret data bits from stego-image pixels, the following steps are accomplished.

Step 1. Divide the stego-image into some nonoverlapping blocks of two consecutive pixels in raster scan manner, say p'_i and p'_{i+1}.

Step 2. Extract secret data bits from LSBs of two successive pixels as follows.

Case 1. If the values of both pixels p'_i and p'_{i+1} are less than 192 (i.e., $p'_i < 192$ and $p'_{i+1} < 192$), extract $k = 3$ bits from k LSB bits of pixel p'_i and again $k = 3$ bits from k LSB bits of pixel p'_{i+1}.

Case 2. If one of the pixel values p_i and p_{i+1} or both of them are larger than or equal to 192 (i.e., $p_i \geq 192$ or $p_{i+1} \geq 192$), extract the secret data bits from two successive pixels of the stego-image as follows.

1. Calculate the difference value d'_i between two consecutive pixels in the block by

$$d'_i = \left|p'_i - p'_{i+1}\right|. \tag{7}$$

2. Apply the range table R_i and obtain the range to which d_i' belongs.
3. Get the number of n secret embedded bits into two pixels p_i' and p_{i+1}' and extract $k = n/2$ secret bits from k LSBs of p_i' and also exploit $k = n/2$ secret bits from k LSBs of p_{i+1}'.

The Experimental Results

In this section, we illustrate the experimental results of the proposed method. In our experiments, we use six 8-bit gray-level images: Elaine, Lena, Peppers, Scene, Zelda, and Tiffany as cover images. The size of all cover images are 512×512, and four of them are shown in Figure 4. The secret data bits are produced by a random number generator. We employ the peak signal-to-noise ratio (PSNR) to evaluate the distortion of the cover images after embedding secret data. If the PSNR value is larger than 30 dB, the distortion of the stego-image is imperceptible to the human eye (Lee et al. 2008). The PSNR value between the cover image P and the stego-image P' can be computed as

$$\text{PSNR} = 10 \times \text{Log}\left(\frac{(255)^2}{\text{MSE}}\right), \tag{8}$$

where MSE is defined by

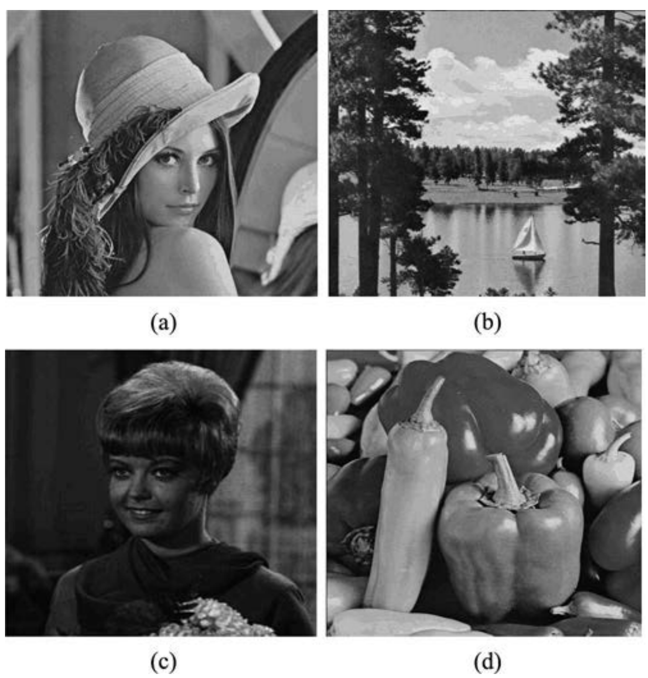

Figure 4. Four 512×512 stego-images obtained by the proposed method: (a) stego-Lena, (b) stego-scene, (c) stego-Zelda, and (d) stego-Peppers.

$$\text{MSE} = \frac{1}{m}\sum_{i=1}^{m}(P_i - P_i')^2, \tag{9}$$

where m is the number of pixels in P and P'. We utilize the universal image quality index to indicate the quality of stego-images (Wang and Bovik 2002). The quality index Q is based on statistic measure and is calculated by

$$Q = \frac{4\sigma_{xy}\bar{x}\bar{y}}{(\sigma_x^2 + \sigma_y^2)[(\bar{x})^2 + (\bar{y})^2]}, \tag{10}$$

where

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N}x_i,\ \bar{y} = \frac{1}{N}\sum_{i=1}^{N}y_i, \tag{11}$$

$$\sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2,$$

$$\sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})^2,$$

$$\sigma_{xy}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y}).$$

Here, N is the size of the cover image, x_i is the value of the pixels in the cover image, y_i is the value of the pixels in the stego-image, $\bar{x}$ is the mean value of x_i, and $\bar{y}$ is the mean value of y_i.

We compare the proposed method results with the results of the adaptive LSB method, which is presented by Yang, Weng, and Wang (2008) and the modulus function method, which is proposed by Lee and Chen (2010). Table 2 indicates that the hiding capacity of the proposed method is better than that of the modulus function and adaptive LSB method, whereas the PSNR values of the proposed method are very near to others.

Table 2. The comparisons of the results between the adaptive LSB method (Yang, Weng, and Wang 2008) and the modulus function method (Lee and Chen 2010).

	Capacity (bit)			PSNR (db)		
Cover Image	Modulus Function	Adaptive LSB	Proposed method	Modulus function	Adaptive LSB	Proposed method
Elaine	786432	800188	837814	37.89	37.56	37.74
Lena	786432	817032	839028	37.91	37.74	37.67
Peppers	786432	807388	822042	37.92	37.05	37.13
Scene	786432	836338	851518	36.28	35.86	36.02
Zelda	786432	789836	799836	37.93	38.12	38.19
Tiffany	786432	811592	834812	37.93	36.68	36.81
Average	786432	810395	830841	37.64	37.17	37.26

Table 3. The comparisons of the results between the adaptive LSB method (Yang, Weng, and Wang 2008) and the modulus function method (Lee and Chen 2010).

Image Quality (Q)			
Cover Image	Modulus Function	Adaptive LSB	Proposed method
Elaine	0.9975	0.9972	0.9973
Lena	0.9976	0.9969	0.9971
Peppers	0.9980	0.9975	0.9977
Scene	0.9985	0.9978	0.9980
Zelda	0.9969	0.9968	0.9969
Tiffany	0.9892	0.9889	0.9891
Average	0.9962	0.9958	0.9960

Also, the PSNR values of the proposed method are above 36 db. So, the distortion of the stego-images resulting in our method is imperceptible.

Table 3 demonstrates that the image quality Q values of the proposed method are better than those of the adaptive LSB method. Moreover, the image quality Q values of our method are perfectly near to the values that are obtained by the modulus function method. Figure 3 shows four stego-images given by the proposed method. The resultant images illustrate that the stego-image quality is high.

Conclusion

In this article, we proposed a new adaptive steganographic method to hide secret data within a gray-scale cover image. First, we divided the cover image into several nonoverlapping blocks with two consecutive pixels and obtained the number of secret bits that could embed into two consecutive pixels. We embedded the secret bits into the cover image by modifying the LSBs of two consecutive pixels. Moreover, we proposed a readjusting procedure to maintain the difference value of the pair of successive pixels in the same range, before and after embedding. Then, we could extract the secret data from the stego-image exactly. The experimental results of our proposed method showed that the quality of the stego-image is acceptable and is better than the method of Yang et al. (2008). Also, the capacity of our method is greater than those of the methods of Yang et al. (2008) and Lee and Chen (2010).

References

Chan, C. K., and L. M. Cheng. 2003. Data hiding in images by simple LSB substitution. *Pattern Recognition* 37 (3):469–74. doi:10.1016/j.patcog.2003.08.007

Chang, K. C., C. P. Chang, P. S. Huang, and T. M. Tu. 2008, June. A novel image steganographic method using tri-way pixel-value differencing. *Journal of Multimedia* 3 (2):37–44. doi:10.4304/jmm.3.2.37-44

Chang, Y. F., and W. L. Tai. 2012. Histogram-based reversible data hiding based on pixel differences with prediction and sorting. *KSII Transaction on Internet and Information Systems* 6 (12):3100–16. doi:10.3837/tiis.2012.12.004

Khodaei, M., and K. Faez. 2010. Image hiding by using genetic algorithm and LSB substitution. In *Proceedings of the international conference on image and signal processing*. Canada: Springer-Verlag.

Lee, C. C., H. C. Wu, C. S. Tsai, and Y. P. Chu. 2008. Adaptive lossless steganographic scheme with centralized difference expansion. *Pattern Recognition* 41:2097–106. doi:10.1016/j.patcog.2007.11.018

Lee, C. F., and H. L. Chen. 2010. A novel data hiding scheme based on modulus function. *Journal of System and Software* 83 (5):832–43. doi:10.1016/j.jss.2009.12.018

Tian, J. 2003. Reversible data embedding using a difference expansion. *IEEE Translation, Circuits and Systems for Video Technology* 13 (8):890–96. doi:10.1109/tcsvt.2003.815962

Wang, R. Z., C. F. Lin, and J. C. Lin. 2001. Image hiding by optimal LSB substitution and genetic algorithm. *Pattern Recognition* 34 (3):671–83. doi:10.1016/s0031-3203(00)00015-7

Wang, Z., and A. C. Bovik. 2002, March. A universal image quality index. *IEEE Signal Processing Letters* 9 (3):81–84. doi:10.1109/97.995823

Wu, D. C., and W. H. Tsai. 2003. A steganographic method for images by pixel-value differencing. *Pattern Recognition* 24 (9–10):1613–26. doi:10.1016/s0167-8655(02)00402-6

Wu, H. C., N. I. Wu, C. S. Tsai, and M. S. Hwang. 2005. Image steganographic scheme based on pixel-value differencing and LSB replacement methods. *IEE Proceedings Vision, Image Signal Processing* 152 (5):611–15. doi:10.1049/ip-vis:20059022

Yang, C. H., C. Y. Weng, H. K. Tso, and S. J. Wang. 2011. A data hiding scheme using the varieties of pixel-value differencing in multimedia images. *Journal of System and Software* 84 (5):669–78. doi:10.1016/j.jss.2010.11.889

Yang, C. H., C. Y. Weng, and S. J. Wang. 2008. Adaptive data hiding in edge areas of images with spatial LSB domain systems. *IEEE Transactions Information Forensics and Security* 3 (3):488–97. doi:10.1109/tifs.2008.926097