A steganographic method for digital images with four-pixel differencing and modified LSB substitution

Xin Liao a,⁎, Qiao-yan Wen a, Jie Zhang b

a State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
b School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, China

Article history:
Received 13 December 2009
Accepted 22 August 2010
Available online 27 August 2010

Keywords:
Steganography
Four-pixel differencing
Modified LSB substitution
Data hiding
Pixel-value differencing
Error block
Embedding capacity
Imperceptible

1. Introduction

Internet is a popular communication channel nowadays. Transmitted data are easy to be copied or destroyed by unauthorized persons. Therefore, how to transmit data secretly by internet becomes an important topic. Encryption may provide a safe way, which transforms data into a ciphertext via cipher algorithms [1]. However, it makes the messages unreadable and suspicious enough to attract eavesdroppers’ attention. To overcome this problem, steganography offers different approaches to transmitting secret messages [2,3]. Steganography is a technique that imperceptibly hides secret data into cover media by altering its most insignificant components for covert communication, such that an unauthorized user will not be aware of the existence of secret data [4].

The most common and well-known steganographic method is called least significant bit (LSB) substitution, which embeds secret data by replacing k LSBs of a pixel with k secret bits directly [5]. Many optimized LSB methods have been proposed to improve this work [6–8]. The human perceptibility has a property that it is sensitive to some changes in the pixels of the smooth areas, while it is not sensitive to changes in the edge areas. Not all pixels in a cover image can tolerate equal amount of changes without causing noticeable distortion. Hence, to improve the quality of stego images, several adaptive methods have been proposed in which the amount of bits to be embedded in each pixel is variable [9–16]. In 2003, Wu and Tsai proposed a novel steganographic method that uses the difference value between two neighboring pixels to determine how many secret bits should be embedded [9]. Chang and Tseng proposed a side match approach to embed secret data, where the number of bits to be embedded in a pixel is decided by the difference between the pixel and its upper and left pixel [10]. In 2005, Wu et al. presented a novel steganographic method, which combined pixel-value differencing and LSB substitution [11]. Park et al. proposed a new method based on the difference value between two pixels adjacent to the target pixel [12]. In 2006, Yang and Weng proposed a multi-pixel differencing method that uses three difference values in a four-pixel block to determine how many secret bits should be embedded [13]. The method in [14] provided the combination of multi-pixel differencing and LSB substitution to improve the work in [13], but the embedding capacity is far less than that of Wu et al.’s method. In 2008, Wang et al. presented a steganographic method that utilizes the remainder of two consecutive pixels to record the information of secret data [15]. Yang et al. proposed an adaptive LSB steganographic method using the difference value of two consecutive pixels to distinguish between edge areas and smooth areas [16]. All pixels are embedded by the k-bit modified LSB substitution method, where k is decided by the range which the difference value belongs to [16].

However, some of them seem not to consider the features of edge sufficiently [9,11,15,16]. The methods in [10,12] have overcome the drawback, but unfortunately they would result in the

© 2010 Elsevier Inc. All rights reserved.
propagated error and lower embedding capacity. In order to provide better stego-image quality and larger embedding capacity, a novel steganographic method improving the multi-pixel differencing based on modified LSB substitution is presented in this paper. Similar to [13,14], a four-pixel block with three difference values is sufficiently considered. The average value of three difference values is exploited to distinguish between edge areas and smooth areas, and to estimate how many secret bits will be embedded into the block. Embed secret bits into each pixel in the block by modified LSB substitution method. Readjustment will be executed to extract secret data exactly and to minimize the perceptual distortion resulted from embedding. The experimental results show that our proposed method provides a large embedding capacity, and the quality of the stego image is improved as well.

The remainder of this paper is organized as follows. In Section 2, the embedding and extracting algorithms of the proposed method is presented. In the next section, we show that the proposed method succeed in embedding and extracting by proving that the readjusting procedure works. The experimental results will be in Section 4. Finally, conclusions are given in Section 5.

2. The proposed method

The proposed method conforms to the issues that are mentioned before. The pixels in edge areas can tolerate much more changes without making perceptible distortion than smooth areas. The range of average difference value is partitioned into two different levels, low level and high level. The division of smooth and edge areas is predefined by users. Pixels located in the block are classified into the block are classified into four-pixel blocks. For instance, assume pixel values \(y_0, y_1, y_2, y_3\) by \((\bar{y}_0, \bar{y}_1, \bar{y}_2, \bar{y}_3)\) in the block, the purpose of 4-bit secret data hiding have been achieved. Repeat Steps 1–6 until all the secret data are embedded in the cover image, and the stego image is obtained.

For example, suppose we have a block with four neighboring pixel values \((139, 140, 140, 154)\) belongs to “Error Block”, because \(D = \frac{y_0 + y_1 + y_2 + y_3}{4} < 6\) and \(154 - 139 = 15 > 2 \times 6 + 2 = 14\). “Error Block” is NOT used to embed secret bits, which will be explained in Section 3.

2.1. The embedding algorithm

All the pixels in the cover image are 256 gray values. The cover image is partitioned into non-overlapping four-pixel blocks. For each block, there are four neighboring pixels \(p_{i-j}, p_{i+j}, p_{j-i}, p_{j+i}\), and their corresponding gray values are \(y_0, y_1, y_2, y_3\), respectively. The detailed embedding steps are as follows.

Step 1: Calculate the average difference value \(D\), which is given by

\[
D = \frac{1}{3} \sum_{i=0}^{1}(y_1 - y_{\text{min}})
\]

\[
y_{\text{min}} = \min\{y_0, y_1, y_2, y_3\}
\]  

(1)

Step 2: Our proposed method adaptively embeds messages using two levels (lower-level and higher-level), and threshold value \(T\) is used to partition the range of \(D\) into two levels. If \(D \leq T\), \(D\) belongs to “lower-level” (i.e., the block belongs to a smooth area), then \(k = k_0\). Otherwise, \(D\) belongs to “higher-level” (i.e., the block belongs to an edge area), then \(k = k_b\). In order to succeed in the readjusting procedure, we apply the restrictions \(2^{k_0} \leq T \leq 2^k\) and \(1 \leq k_b, k_0 \leq 5\).

Step 3: Verify whether the block belongs to “Error Block”. If not, continue to next step. Otherwise, restart from Step 1.

Definition 1. Assume \(y_{\text{max}} = \max\{y_0, y_1, y_2, y_3\}\), the block is called “Error Block” if and only if \(D \leq T\) and \(y_{\text{max}} - y_{\text{min}} > 2T + 2\).

For instance, assume \(T = 6\). A block with four neighboring pixel values \((139, 140, 140, 154)\) belongs to “Error Block”, because \(D = \frac{y_0 + y_1 + y_2 + y_3}{4} < 6\) and \(154 - 139 = 15 > 2 \times 6 + 2 = 14\). “Error Block” is NOT used to embed secret bits, which will be explained in Section 3.

Step 4: Convert \(y_i\) to be \(y'_i\) by the \(k\)-bit common LSB substitution method \((0 \leq i \leq 3)\), respectively.

Step 5: Apply the \(k\)-bit modified LSB substitution method to \(y'_i\), and let \(y''_i\) be the result \((0 \leq i \leq 3)\), respectively.

Step 6: This step is called “readjusting procedure”. Let \(\tilde{y}_i = y''_i + I \times 2^3, 0 \leq I \leq 3, I \in \{0, 1, -1\}\), and search \((\tilde{y}_0, \tilde{y}_1, \tilde{y}_2, \tilde{y}_3)\) such that

1. \(D\) and \(D\) belong to the same level, where \(D = \frac{1}{3} \sum_{i=0}^{1}(\tilde{y}_1 - y_{\text{min}})\), \(y_{\text{min}} = \min\{\tilde{y}_0, \tilde{y}_1, \tilde{y}_2, \tilde{y}_3\}\).

2. The final stego block \((\tilde{y}_0, \tilde{y}_1, \tilde{y}_2, \tilde{y}_3)\) does not belong to “Error Block”.

3. The value of \(\sum_{i=0}^{1}(\tilde{y}_i - y_{\text{min}})^2\) is minimized.

After the replacement of \((y_0, y_1, y_2, y_3)\) by \((\tilde{y}_0, \tilde{y}_1, \tilde{y}_2, \tilde{y}_3)\) in the block, the purpose of 4-bit secret data hiding have been achieved. Repeat Steps 1–6 until all the secret data are embedded in the cover image, and the stego image is obtained.

For example, suppose we have a block with four neighboring pixel values \((139, 146, 137, 142)\), and the secret data are \(001111111110\). Assume \(T = 5\), \(k_b = 2\) and \(k_0 = 3\). Calculate the average difference value \(D = \frac{y_0 + y_1 + y_2 + y_3}{4} > T = 5\), then \(y_i(0 \leq i \leq 3)\) are embedded by the 3-bit common LSB substitution method at first, \(y_0 = 136, y_1 = 151, y_2 = 143\) and \(y_3 = 141\). After applying the 3-bit modified LSB substitution method, \(y'_0 = 136, y'_1 = 143, y'_2 = 135\) and \(y'_3 = 141\). Readjustment is executed resulting in \(\tilde{y}_0 = 136, \tilde{y}_1 = 151, \tilde{y}_2 = 135\) and \(\tilde{y}_3 = 141\).

2.2. The extracting algorithm

In the extraction process, we can quickly extract secret data without the original image. Partition the stego image into four-pixel blocks, which is identical with the embedding algorithm. For each block \((p_{i-j}, p_{i+j}, p_{j-i}, p_{j+i})\), the following steps are executed to extract the secret data.

Step 1: Calculate the average difference value \(D\) by Eq. (1).

Step 2: Use the threshold value \(T\) to find out the level which \(D\) belongs to. If \(D\) belongs to the “lower-level”, \(k = k_0\), otherwise \(k = k_b\).

Step 3: Verify whether the block belongs to “Error Block”. If not, extract 4-bit secret data from the k-bit LSB of \(y_i(0 \leq i \leq 3)\). Otherwise, restart from Step 1.

For instance, we extract the embedding example \((136, 151, 135, 141)\), which is shown in the above subsection. Since \(T = 5\), \(D = \frac{y_0 + y_1 + y_2 + y_3}{4}\) belongs to “higher-level”, then \(k = k_b = 3\). We extract 3-bit LSB of \(y_i(0 \leq i \leq 3)\), respectively. Secret data \(0011111110\) can be obtained.

3. Theoretic analyses and discussions

Secret data can be directly extracted as the least \(k\) bits of the pixel values, because Step 4 and Step 5 in the embedding algorithm...
Lemma 1 [16]. For the k-bit modified LSB substitution, if pixel value $y_i$ dose not belong to the rang $[0,2^{k-1}]$ or $[255 - 2^{k-1},255]$, $y_i$ is modifiable and the resulting value $y_i'$ such that $y_i - 2^{k-1} \leq y_i' \leq y_i + 2^{k-1}$.

Lemma 2 [16]. For the k-bit modified LSB substitution, if $y_i$ belongs to the rang $[0,2^{k-1}]$, the embedded result has $y_i - 2^{k-1} \leq y_i' \leq y_i + (2^{k-1} - 1)$ and $y_i'$ belongs to the rang $[0,2^{k-1}]$; if $y_i$ belongs to the rang $[255 - 2^{k-1},255]$, the embedded result has $y_i - (2^{k-1} - 1) \leq y_i' \leq y_i + 2^{k-1}$ and $y_i'$ belongs to the rang $[255 - (2^{k-1} - 1),255]$.

Assume $y_i' - y_i = \Delta y_i$, $y_i' - y_i = \Delta y_i$, we have $y_0 \leq y_1 \leq y_0 \geq y_i'$. Then, $0 \leq y_1 - y_0 \leq y_0 - y_2$.

$y_0' + 3y_2 - 3y_0 - y_1 = (y_0 + \Delta y_0) + 3y_2 - 3(y_1 + \Delta y_1) - y_1$

$= -4(y_1 - y_0) + \Delta y_0 - \Delta y_1 y_0 \leq \Delta y_0 - 3\Delta y_1 \leq 2^{k-1}$

$y_0' + 3y_2 - 3y_0 - y_1 = -4(y_1 - y_0) + \Delta y_0 - 3\Delta y_1$

$\geq -4\Delta y_0 - \Delta y_1 + \Delta y_0 - 3\Delta y_1$

$\geq -3\Delta y_0 + y_1 \geq -2^{k-1}$

Thus, we have $|y_0' + 3y_2 - 3y_0 - y_1| \leq 2^{k-1}$. From Eq. (3) we know that $|D' - D| \leq \frac{1}{2} \times 2^{k-1} + \frac{1}{2} \times 2^{k-1} = 2^k$. □

Theorem 2. Suppose that four pixels of the block are modifiable and the cover block does not belong to "Error Block", then the readjusting phase can work successfully when $2^k \leq T \leq 2^k$ and $1 \leq k_s k_b \leq 5$.

Proof. There are eighty-one readjusting decisions $(3^4 = 81)$ in Step 6, and we will show that there exists at least one choice such that the former two conditions hold in Step 6; namely, (1) $D$ and $D'$ belong to the same level; (2) The final step block does not belong to “Error Block”. Then, there must exist the best one in the finite choices such that three conditions hold in Step 6. Thus, the readjusting phase works successfully.

Without loss of generality, suppose that the resulting pixel values are $w < z < y < x$ after the k-bit modified LSB substitution. From Theorem 1, we have $D' = \frac{1}{2}(x + y + z - 3w)$ and $|D' - D| \leq 2^k$. Now we discuss the cases that $D$ belongs to different levels, which can be divided into two categories:

(1) Category A: $D > T$ belongs to high level, $k = k_h$.

Case 1: If $D' > T$, there already exists a choice $(x,y,z,w)$ such that the former two conditions hold in Step 6.

Case 2: If $D' \leq T$, a choice $(x,y,z,w = 2^k)$ such that $D = D' + 2^k = D' + 2^k \geq T$. Decrease of $w$ by $2^k$ may not be allowed if the pixel value $w < 2^k$. However, we can find another choice $(x - 2^k, y + 2^k, z + 2^k, w)$ such that $D = D' + 2^k \geq T$ if $z \leq y < x < 255 - 2^k$. Obviously, these choices don’t belong to “Error Block”.

(2) Category B: $D \leq T$ belongs to low level, $k = k_l$. The cover block does not belong to “Error block”, so we have $y_{\text{max}} - y_{\text{min}} \leq 2T + 2$, then $x - w \leq (y_{\text{max}} + 2^k) - (y_{\text{min}} + 2^k - 2T - 2^{k-1}) \leq 2T + 2 + 2^{k-1}$.

Case 1: $D' \leq T$. If $(x,y,z,w)$ does not belong to “Error block”, it satisfies the former two conditions in Step 6. Otherwise, $x - w > 2 \times T + 2$, we can find another choice $(x - 2^k, y, z, w)$ such that the former two conditions hold.

Case 2: $D' > T$. We divide it into 10 subcases according to the relationship among four pixels $x, y, z, w$, and as follows:

subcase 1: $x \geq y \geq z \geq w + 2^k \times 2 > w$. Fig. 1(a) shows the possible location. There exists a choice $(x, y, z, w + 2^k)$ such that $D = D' - 2^k < D < T$ and $x - (w + 2^k) \geq 2T + 2$. Fig. 1(b) shows the possible location. There exists a choice $(x, y, z, w + 2^k)$ such that $D = D' + 2^k > D < T$ and $x - (w + 2^k) \leq 2T$. Fig. 1(c) shows the possible location. There exists a choice $(x, y, z, w + 2^k)$ such that $D = D' - 2^k < D < T$ and $x - (w + 2^k) \geq 2T + 2$. Fig. 1(d) shows the possible location.
(x,y,z,w+2^k) such that \( \hat{D} = D^* - 2^k \leq T \) and \( x-(w+2^k) \leq 2T+2 \).

Subcase 5: \( x \geq w+2^k \times 2 \times y \geq w+2^k > z \geq w \). Fig. 1(e) shows the possible location. There exists a choice \((x-2^k,y,z+w+2^k)\) such that \( \hat{D} = D^* - 2^k \leq D \leq T \) and \( \max(x-2^k,y,z+2^k) - (w+2^k) \leq 2T+2 \).

Subcase 6: \( x \geq w+2^k \times 2 \times w+2^k > y \geq z \geq w \). Fig. 1(f) shows the possible location. There exists a choice \((x-2^k,y+2^k,z+w+2^k)\) such that

\[
\hat{D} = \frac{1}{3} \times \left[ (x-2^k-(w+2^k)) + (y+2^k-(w+2^k)) \right] = \frac{1}{3} \times \left[ (x-w) + (y-w) + (z-w) \right] \\
\frac{-2}{3} \times 2^k \leq \frac{1}{3} \times \left[ (2T+2^k + (2^k-1) + (2^k-1) \right] \\
\frac{-2}{3} \times 2^k \leq \frac{1}{3} \times (2T+2^k) \leq T
\]

and \( \max(x-2^k,y+2^k,z+2^k) - (w+2^k) \leq 2T+2 \).

Subcase 7: \( w+2^k \times 2 \times x \geq y \geq z \geq w+2^k > w \). Fig. 1(g) shows the possible location. There exists a choice \((x,y,z,w+2^k)\) such that \( \hat{D} = D^* - 2^k \leq D \leq T \) and \( x-(w+2^k) \leq 2T+2 \).

Subcase 8: \( w+2^k \times 2 \times x \geq y \geq w+2^k > z \geq w \). Fig. 1(h) shows the possible location. There exists a choice \((x-2^k,y,z,w)\) such that \( \hat{D} = D^* - 2^k \leq D \leq T \) and \( \max(x-2^k,z) - w \leq 2T+2 \).

Subcase 9: \( w+2^k \times 2 \times x \geq w+2^k > y \geq z \geq w \). Fig. 1(i) shows the possible location. There exists a choice \((x-2^k,y,z,w)\) such that \( \hat{D} = D^* - 2^k \leq D \leq T \) and \( \max(x-2^k,z) - w \leq 2T+2 \).

Subcase 10: \( w+2^k \times x \geq y \geq z \geq w \). Fig. 1(j) shows the possible location. The subcase 10 does not belong to Case 2 because \( D^* < 2^k \times (2^k+2^k) = 2^k \leq T \), which contradicts the assumption imposed for Case 2. \( \square \)

Theorem 2 has shown that our method is correct if all the four pixels are modifiable. Now, we discuss the cases that at least one of the pixels is not modifiable.

**Lemma 5.** If there exist a pixel belongs to the range \([0,2^k-1]\) and another pixel belongs to the range \([255-2^k,255]\), there already exists a choice such that the former two conditions hold in Step 6.

**Proof.** Suppose the pixels \( 0 \leq p_1 \leq 2^k-1 \) and \( 255-2^k-1 \leq p_2 \leq 255 \), then \( D \geq \frac{1}{4} \times (255-2^k-1-2^k) > 2^k \geq T \) and \( D \) belongs to high level. From Lemma 2, we have \( 0 \leq p_1 \leq 2^k-1 \) and \( 255-2^k-1 \leq p_2 \leq 255 \). \( D' > \frac{1}{4} \times (255-2^k-1-2^k-1) > 2^k \geq T \) and \( D' \) also belongs to high level. Obviously, the choice does not belong to “Error Block”. \( \square \)

**Theorem 3.** Suppose that not all the four pixels of the block are modifiable and the cover block does not belong to “Error Block”. Then, the readjusting phase can work successfully when \( 2^k \leq T \leq 2^{k+1} \) and \( k \leq k_l, k_h \leq 5 \).

**Proof.** Similar to Theorem 2, we will show that there exists at least one choice such that the former two conditions hold in Step 6. It can be divided into three cases: (1) Only some pixels belong to \([0,2^k-1]\); (2) Only some pixels belong to \([255-2^k,255]\); (3) Some pixels belong to \([0,2^k-1]\), and some belong to \([255-2^k-1,255]\). We only need to discuss case 1, because the proof of case 2 is similar and case 3 is solved by Lemma 5. We divide case 1 into two categories:

(1) **Category A:** \( D > T \) belongs to high level. No matter how many pixels are not modifiable in the four-pixel block, the proof in Theorem 2 is still efficient and correct.

(2) **Category B:** \( D \leq T \) belongs to low level.

Case 1: \( D' \leq T \). The cover block is not an error block, so we have \( y_{max} - y_{min} \leq 2T+2 \). If \( (x,y,z,w) \) does not belong to “Error block”, it satisfies the former two conditions in Step 6. Otherwise, \( x-w \geq 2 \times y+z \), we have to consider whether \( y_{max} \) is modifiable or not. If \( y_{max} \) is modifiable, then \( (x-w) \leq (y_{max} - 2^k) - (y_{min} - 2^k) \leq 2T+2 \), we can find another choice \((x-2^k,y,z,w)\) such that the former two conditions hold in Step 6. If \( y_{max} \) is not modifiable, i.e., \( y_{max} \) belongs to \([0,2^k-1]\). Thus, all the cover four pixels are not modifiable.
From Lemma 4 we know that $x - w \leq 2k^2 + 2T + 2$, there exists a choice $(x, y, z, w)$ such that the former two conditions hold. 

Case 2: $D > T$. We also divide it into 10 subcases according to the relationship among four pixels $x, y, z, w$.

Only some pixels in the four-pixel block belong to $[0, 2k^1]$ and there are four conditions:

Condition 1: Only one pixel $y_0$ belongs to $[0, 2k^1]$, and others are modifiable. All the subcases in Fig. 1 are possible and $y_0 = \min(y_{01}, y_{02}, y_{11}, y_{12})$. The block is not an error block, so we have $y_{\text{max}} - y_{\text{min}} < 2T + 2$, then $x - w = (y_{\text{max}} + 2k^1) - (y_{\text{min}} - 2k^1) \leq 2T + 2 + 2k^2$.

For the subcases 1, 2, 4, 7: From Lemma 2, we have $0 \leq y_0 < 2k^1 - 1$. Thus, $y_0 = w$. From Lemma 3, we know that $D^1 = D = \frac{1}{2}(|y_0 - y_0| - |y_1 - y_0|) + (y_0^2 - y_0) - y_2 - y_0).$ Then $(x, y, z, w + 2k^2)$ is a choice such that $D = D^1 = x + 2k^2 < T$ and $x - w = 2(2k^2 + 2T + 2) + 2k^2$.

For the subcases 3, 5: If $y_0 = w$, we also have $D^2 = D < 2k^2$ and $(x, y, z, w + 2k^2)$ is a choice such that $D = D^2 < 2k^2 < T$ and $x - w = 2T + 2$. If $y_0 \neq w$, we know that $y_0 = z$ according to Lemma 2. Suppose $y'_0 = w$, we have $D^1 = D = \frac{1}{3}(y'_0 - y'_0 + y'_1 - 3y'_1) - \frac{1}{3}(y'_1 + y'_2 + y'_3 - 3y'_0) = \frac{1}{3}(y'_0 - y'_0 + y'_1 - 3y'_1) + \frac{1}{3}(y'_0 + y'_3 - y'_1) - \frac{1}{3}(y'_1 + y'_2 + y'_3 - 3y'_0)$

$$= \frac{1}{3}(y'_0 - y'_0 + y'_1 - 3y'_1) + \frac{1}{3}(y'_0 + y'_3 - y'_1) + \frac{1}{3}(z - w) + 2(y_1 - y'_1)]$$

$$\leq \frac{1}{3}(y'_0 - y'_0 + y'_1 - 3y'_1) + \frac{1}{3}(y'_0 + y'_3 - y'_1) + \frac{1}{3}(z - w) + 2(y_1 - y'_1)]$$

$$\leq \frac{1}{3}y'_0 - \frac{1}{3}y'_0 + \frac{1}{3}y'_1 - \frac{1}{3}(y'_1 + y'_2 + y'_3 - 3y'_0)$$

$$\leq \frac{1}{3}y'_0 - \frac{1}{3}y'_0 + \frac{1}{3}y'_1 - \frac{1}{3}(y'_1 + y'_2 + y'_3 - 3y'_0)$$

$$(x, y, z, w + 2k^2) = \frac{1}{3}y'_0 - \frac{1}{3}y'_0 + \frac{1}{3}y'_1 - \frac{1}{3}(y'_1 + y'_2 + y'_3 - 3y'_0) + \frac{1}{3}(y'_0 - y'_0 + y'_1 - 3y'_1) = 2k^2$$

Then $(x, y, z, w + 2k^2)$ is a choice such that $D^1 = D < 2k^2 < T$ and $x - w = 2T + 2 + 2k^2$.

For the subcases 6, 8, 9, 10: The proof in Theorem 2 is still efficient and correct.

Condition 2: Two pixels $y_0$ and $y_1$ belong to $[0, 2k^1]$, and others are modifiable. From Lemma 2, we know that $0 \leq y'_0 < y'_1 < 2k^1 - 1$, so the subcases 1, 2, 4, 7 are impossible.

For the subcases 3, 5: Because $0 \leq y'_0 < y'_1 < 2k^1 - 1$, $y'_0 = w$ or $y'_1 = w$. If $y'_0 = w$, from Lemmas 3 and 4 we have $D^1 = D = \frac{1}{2}(y'_0 - y'_0) - y_2 - y_0) + (y'_0 - y'_0) - y_2 - y_0).$ Then $(x, y, z, w + 2k^2)$ is a choice such that $D = D^1 = x + 2k^2 < T$ and $x - w = 2T + 2 + 2k^2$. If $y'_0 = w$, we have $D^1 = D = \frac{1}{3}(y'_0 + y'_2 + y'_3 - 3y'_1) - \frac{1}{3}(y'_1 + y'_2 + y'_3 - 3y'_0) = \frac{1}{3}(y'_0 + y'_2 + y'_3 - 3y'_1) + \frac{1}{3}(y'_0 + y'_3 - y'_1) + \frac{1}{3}(y'_0 + y'_3 - y'_1) + \frac{1}{3}(y'_0 + y'_3 - y'_1) + \frac{1}{3}(z - w) + 2(y_1 - y'_1)]$
shows the results of the proposed method in terms of embedding capacity and PSNR value. The embedding capacities (in bits) and the PSNR values are average values of the results executed by random bit streams many times.

Table 3 shows the comparisons of the results between Wu et al.’s[11] and ours in terms of embedding capacity and PSNR value, where the proposed method uses 3–4 division with $T = 15$. It is shown that our method provides both much larger embedding capacity and better image quality.

A number of secret bits are embedded into every block using the modified LSB substitution method, which was used by Yang et al.’s[16]. Parameters $k_l/C_0$ and $k_h$ both used in Yang et al.’s and ours represent the same condition. A steganographic scheme is preferred if it can provide higher PSNR values when concealing with the same embedding capacity. Thus, the comparisons of PSNR values based on the embedding capacities that are slightly larger than those provided by Yang et al. are given in Table 4. The statistics show that the proposed steganographic method is better than Yang et al.’s approach when using a two-level adaptive embedding strategy. A possible reason is that our proposed method considers non-overlapping 2\texttimes 2 pixel blocks instead of two consecutive pixels, so the features of edge may be considered sufficiently and the pixels in edge areas can tolerate much more changes without making perceptible distortion.

5. Conclusions

In this paper, we have proposed a novel steganographic method based on four-pixel differencing and modified LSB substitution. Secret data are hidden into each pixel by the $k$-bit modified LSB substitution method, where $k$ is decided by the average difference value of a four-pixel block. Redjustment has been executed to extract the secret data exactly and to minimize the perceptual distortion.
The lemmas and theorems presented in Section 3 are of theoretical importance, because the correctness and feasibility of the proposed method are validated explicitly. The proposed method considers the features of edge sufficiently, so the pixels in edge areas can tolerate much more changes without making perceptible distortion. The experimental results show that the proposed method provides larger embedding capacity and better image quality. Our proposed method majors in more significant promotions in the terms of capacity and imperceptivity. There is a trade-off between embedding capacity/quality and attack-resistance, and it would sacrifices attack-resistance a little for obtaining higher embedding capacity/quality. In the future, besides the merits achieved in this paper, we will attempt to improve and modify the propose method to achieve stronger security.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant Nos. 60873191, 60903152, 60821001), Beijing Natural Science Foundation (Grant No. 4072020).

<table>
<thead>
<tr>
<th>Covers</th>
<th>T = 7, 2–3</th>
<th>T = 12, 2–4</th>
<th>T = 15, 3–4</th>
<th>T = 18, 2–5</th>
<th>T = 18, 3–5</th>
<th>T = 21, 4–5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Capacity</td>
<td>PSNR</td>
<td>Capacity</td>
<td>PSNR</td>
<td>Capacity</td>
<td>PSNR</td>
</tr>
<tr>
<td>Elaine</td>
<td>668384</td>
<td>42.19</td>
<td>648040</td>
<td>39.71</td>
<td>821640</td>
<td>38.98</td>
</tr>
<tr>
<td>Lena</td>
<td>587220</td>
<td>44.05</td>
<td>590088</td>
<td>41.87</td>
<td>810564</td>
<td>39.57</td>
</tr>
<tr>
<td>Baboon</td>
<td>714164</td>
<td>41.57</td>
<td>803656</td>
<td>37.09</td>
<td>903580</td>
<td>36.90</td>
</tr>
<tr>
<td>Peppers</td>
<td>587624</td>
<td>44.28</td>
<td>576416</td>
<td>42.49</td>
<td>805492</td>
<td>39.79</td>
</tr>
<tr>
<td>Toys</td>
<td>584932</td>
<td>44.15</td>
<td>600488</td>
<td>41.16</td>
<td>816852</td>
<td>39.36</td>
</tr>
<tr>
<td>Girl</td>
<td>599736</td>
<td>43.69</td>
<td>591832</td>
<td>41.73</td>
<td>808584</td>
<td>39.61</td>
</tr>
<tr>
<td>Gold</td>
<td>624856</td>
<td>43.08</td>
<td>617696</td>
<td>40.80</td>
<td>816892</td>
<td>39.27</td>
</tr>
<tr>
<td>Barb</td>
<td>658560</td>
<td>42.52</td>
<td>724208</td>
<td>38.37</td>
<td>871184</td>
<td>37.67</td>
</tr>
<tr>
<td>Zelda</td>
<td>564248</td>
<td>44.73</td>
<td>557832</td>
<td>43.48</td>
<td>797268</td>
<td>40.16</td>
</tr>
<tr>
<td>Tiffany</td>
<td>579948</td>
<td>44.25</td>
<td>579952</td>
<td>42.32</td>
<td>805760</td>
<td>39.77</td>
</tr>
<tr>
<td>Average</td>
<td>616067</td>
<td>41.45</td>
<td>628985</td>
<td>40.94</td>
<td>825782</td>
<td>39.11</td>
</tr>
</tbody>
</table>

This work is supported by National Natural Science Foundation of China (Grant Nos. 60873191, 60903152, 60821001), Beijing Natural Science Foundation (Grant No. 4072020).
References