Image Steganography Based on Hamming Code and Edge Detection

Shuliang Sun
School of Electronics and Information Engineering, Fuqing Branch of Fujian Normal University, China

Abstract: In this paper a novel algorithm which is based on hamming code and $2^k$ correction is proposed. The new method also utilizes canny edge detection and coherent bit length. Firstly, canny edge detector is applied to detect the edge of cover image and only edge pixels are selected for embedding payload. In order to enhance security, the edge pixels are scrambled. Then hamming encoding is practiced to code the secret data before embedded. Calculate coherent bit length $L$ on the base of relevant edge pixels and replace with $L$ bits of payload message. Finally, the method of $2^k$ correction is applied to achieve better imperceptibility in stego image. The experiment shows that the proposed method is more advantage in PSNR, capacity and universal image quality index ($Q$) than other methods.

Keywords: hamming code, $2^k$ correction, coherent bit length, canny edge detector

1. Introduction

With the rapid development in both communication and Internet technologies, digital signal can be transmitted conveniently over the Internet. However, Internet is an open transmission medium, communication over the Internet will bring us not only convenience but also some hazards and risks. It is more and more important to protect information to communicate securely nowadays. There are two important techniques to protect secret data: cryptography and information hiding. As a kind of information hiding technology, steganography has been developed rapidly and caused widespread attention [13]. Steganography is a skill that hides the existence of secret message under the cover of a carrier signal. As a result, no one apart from the sender and intended recipient realizes there is a hidden message.

The original image is called the cover image, which is used to carry secret message as a carrier signal. The secret message is embedded into cover medium. The image which has carried secret message is called the stego image. The most important features in steganographic system are embedding payload, security (imperceptibility) and robustness. A steganographic scheme with low image distortion, high payload and security is more expected. However, the three factors are inevitable conflict. Therefore, there is a compromise among them depending on different application requirements [14].

Image steganography is widely used over the internet. In order to increase safety, secret information was encoded before embedded in the cover image. Since human eyes are more sensitive to smooth areas than edge regions, secret information will be embedded in the edge pixels of cover image [3, 4].

The rest of the paper is organized as follows: The related work is displayed in section 2. Related skills are discussed in section 3. The proposed algorithms of embedding and extraction are presented in section 4. Experiments and conclusion are showed in section 5 and 6.

2. Related Works

Bassil proposed an image steganography algorithm based on the canny edge detection [3]. Secret data was hidden within boundary pixels of the cover image.

Three parameters were provided in the algorithm and three LSBs of every color edge pixel were replaced with bits of secret data. In Jain et al. [7] proposed a way to search the edges of the cover image
that could be used to hide secret message. It also provided the depth view of image steganography and edge detection filter techniques. In [19], Yu et al proposed a new steganography scheme which was based on the methods of just noticeable difference and contrast sensitivity function. The method of 2^k correction was also applied to get better imperceptibility. The steganography technique could provide more payloads and had better imperceptibility in [10]. In [15], a novel secure steganography method was put forward. It was based on edge detection and Huffman Encoding. Coherent bit length was also adopted to embed different bits of secret data according to the values of edge pixels values. Authors in [4, 11, 17, 20] represented a steganography technique based on Hamming Code. Secret data was encrypted with Hill cipher before embedding in the cover image [9, 12]. Other coding methods were also used in steganography technique [1, 2, 5, 8, 16].

3. Proposed Work

3.1. Hamming Code

Hamming code is proposed by Richard Hamming [6]. It is a block code which could detect and correct error.

Especially, (7, 4) hamming code is the most widely used. Four data bits (m_1, m_2, m_3, and m_4) are encoded by adding three parity bits (p_1, p_2, and p_3) to become the codeword of length 7. Each parity check bit is created by its associated data bits. The hamming code detects errors by ensuring each parity check bit and its corresponding data bits achieve the even parity [11].

This detection procedure is called parity checking.

\[ 2^r - 1 \geq n \quad \text{or} \quad 2^r - 1 \geq k + r \quad r \geq 2 \]  \hspace{1cm} (1)

Where n is the length of codeword, k is the length of secret data and r is the length of parity bits.

Table 1 illustrates the relationship between parity check bits and data bits, where “√” indicates the relationship exists between parity check bits and associated data bits.

<table>
<thead>
<tr>
<th>m_1</th>
<th>m_2</th>
<th>m_3</th>
<th>m_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>p_1</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>p_2</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>p_3</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

As shown in table 1, the parity check bits can be created by using the associated date bits. The results are shown in Table 2, where “⊕” donates an exclusive or (XOR) operation.

\[ p_1 = m_1 \oplus m_2 \oplus m_4 \]  \hspace{1cm} (2)
\[ p_2 = m_1 \oplus m_3 \oplus m_4 \]  \hspace{1cm} (3)
\[ p_3 = m_2 \oplus m_3 \oplus m_4 \]  \hspace{1cm} (4)

<table>
<thead>
<tr>
<th>No.</th>
<th>codeword</th>
<th>No.</th>
<th>codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>m_1</td>
<td>m_2</td>
<td>m_3</td>
<td>m_4</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The way to combine data and parity together is to put the parity bits at position 2^r such as p_1, p_2, m_1, p_3, m_2, m_3, m_4.

Parity matrix H and generator matrix G are both applied in hamming code. Especially matrix G is used for encoding and matrix H is used for decoding. On the sending side, each 4-bit data will be multiplied by the matrix G and the result is modulo of 2. The codeword X will be obtained and sent through the communication channel.

\[ X = M \times G \]  \hspace{1cm} (5)

Where \[ M = (m_1, m_2, m_3, m_4) \] and

\[ G = \begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 1
\end{bmatrix} \]

At the destination, in order to check the encoded message of 7-bit R (data and parity) that has been received, matrix R will be multiplied by the transpose of the parity check matrix H, and the result is modulo of 2 again.

\[ Z = R \times H^T \]  \hspace{1cm} (6)

Where
The vector $Z$ is composed of three bits $(z_1, z_2, z_3)$. The vector $Z$ will be all zeroes $(000)$ if the message is correct during transmission. Otherwise, any change in the message will lead to flipping one or more bits of the message; then it needs an error correction process.

### 3.2. Coherent Bit Length $L$

The basic idea of embedding is to replace the least significant bits (LSB) of the cover image with the most significant bits (MSB) of the secret image. It should not undermine the visual effects of the cover image obviously after embedding. Different pixel value could be embedded different number of binary bits. Coherent bit length $L$ determines the payload of each edge pixels $(E_i)$. The parameter $L$ is calculated according to the conditions given below [15]:

- If $E_i \geq 2^7$; $L=4$
- If $2^6 \leq E_i < 2^7$; $L=3$
- If $2^5 \leq E_i < 2^6$; $L=2$
- Else $L=1$

The conditions to determine $L$ also serves as secret key to retrieve the payload at the destination.

### 3.3. $2^k$ correction method

A mathematic method is used to achieve better imperceptibility in stego image. In some cases there are some differences between cover pixel and stego pixel after embedding. To over come these differences and get better vision effects, $2^k$ correction method is adopted [12].

The process of $2^k$ correction is defined as follows:

- Error value $(EV) = |$Actual pixel value $(APV) - $Stego pixel value $(SPV)|$,

- Parameter $k$ is the number of bits which are embedded in an edge pixel value.

If $(SPV-APV) \geq 2^{k-1}$ & $(SPV-2^k) \leq 0$
- New stego pixel value = $SPV + 2^k$

Else if $(SPV-APV) < -2^{k-1}$ & $(SPV+2^k) < 255$
- New stego pixel value = $SPV + 2^k$

Else
- New stego pixel value = $SPV$

In this way, the $2^k$ correction makes the new stego pixel value closer to the actual pixel value $(APV)$ without affecting the secret data.

### 4. The Proposed Algorithm

#### 4.1. Data Embedding Algorithm

Algorithm of the embedding procedure

1) Read the cover image and secret image.
2) Canny edge detector is executed to detect the boundaries of cover image.
3) Scramble the edge pixels with key 1.
4) Secret image is converted to 1-D bit streams.
5) Scramble the 1-D bit streams with key 2.
6) Calculate coherent bit length $L$ according to the value of edge pixel.
7) Encode bit streams using Hamming $(7, 4)$ encoder and the result of the encoded data is XORed with the 7 bits of random value using key3.
8) Binary bit streams, coherent bit length $L$ are embedded into scrambling edge pixels.
9) Unscramble edge pixels.
10) The method of $2^k$ correction is applied to achieve better imperceptibility in stego image.
11) New stego image is obtained.

#### 4.2. Data Extraction Algorithm

Algorithm of extracting procedure

1) Get the stego image.
2) Perform $2^k$ correction method on the stego image.
3) Canny edge detector is performed to detect the edge of stego image.
4) Rearrange edge pixels with key1 and obtain original sequence.
5) Calculate the coherent bit length L.
6) Extract L bits LSBs of edge pixels.
7) The message is XORed with random numbers.
8) Decode secret message using hamming (7, 4) decoder.
9) Reposition the whole message into the original order with key2.
10) Convert the bit streams to 2-D array.
11) Secret image is achieved.

Especially, key1, key2 and key3 are generated by pseudorandom number generator.

5. Experiments

In this paper, the experiment is simulated using MATLAB 10 program on Windows 7. To measure image quality of the proposed scheme, Peak Signal-to-Noise Ratio (PSNR), Universal Image Quality Index (Q) and Capacity are adopted. To calculate PSNR, first MSE (Mean Square Error) is calculated using Equation 7:

\[
MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} ||I(i, j) - S(i, j)||^2
\]  

(7)

MSE is the cumulative squared error between original image (I) and stego image (S). M and N are the size of row and column of cover image. Thereafter PSNR value is calculated using equation 8 in decibels. A higher value of PSNR is better because of the superiority of the signal to that of the noise.

\[
PSNR = 20 \log_{10} \left( \frac{255}{\sqrt{MSE}} \right)
\]  

(8)

Universal image quality index (Q) was developed by Wang [18]. Image distortion is often evaluated with three factors: loss of correlation, luminance distortion and contrast distortion. Traditional methods use error summation to measure image quality, but image quality measure Q is proposed by combining three factors which are employed to depict image distortion.

\[
Q = \frac{4\sigma_{xy} \bar{x} \bar{y}}{\sigma^2_x + \sigma^2_y + [(\bar{x})^2 + (\bar{y})^2]}
\]  

(9)

Where \( \bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \), \( \bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i \)

\[
\sigma^2_x = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2
\]  

\[
\sigma^2_y = \frac{1}{N-1} \sum_{i=1}^{N} (y_i - \bar{y})^2
\]

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

\( x = \{x_i | i = 1, 2, \ldots, N\}, y = \{y_i | i = 1, 2, \ldots, N\} \)

Capacity means the number of bits in a cover image that can be modified without deteriorating the vision effect. The embedding capacity demands to preserve the statistical properties of the cover image. It is represented by bits per pixel (bpp).

The ways of LSBs and Jae-Gil Yu’s [18] are compared with proposed method in this paper. Cover images of 512x512 gray-scale were used in the experiments. Secret image is an 8-bit grayscale image of size 256x256.

Figure 3 shows the results of applying the canny edge detection algorithm on the input image. As for the parameters, the size of the Gaussian filter is set to 1.5; the low threshold is arranged to 5, and the high threshold is appointed to 40.

Figure 3. The edge images are generated by the canny edge detector.

In this paper, secret data is only embedded in edge pixels in three methods. Though payload is smaller in this paper than with traditional methods, the values of PSNR and Q are much larger than other techniques. From table 3, it can be concluded that the proposed technique is better than LSB-3 and Jae-Gil Yu’s in three parameters.
Table 3. Comparison of Capacity, PSNR and Q.

<table>
<thead>
<tr>
<th>Cover images</th>
<th>LSB-3</th>
<th>Jae-Gil Yu’s</th>
<th>Proposed Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>Capacity (bit)</td>
<td>36985</td>
<td>37348</td>
</tr>
<tr>
<td></td>
<td>PSNR (dB)</td>
<td>51.2655</td>
<td>55.5273</td>
</tr>
<tr>
<td></td>
<td>Q</td>
<td>0.9826</td>
<td>0.9925</td>
</tr>
<tr>
<td>Peppers</td>
<td>Capacity (bit)</td>
<td>33205</td>
<td>34263</td>
</tr>
<tr>
<td></td>
<td>PSNR (dB)</td>
<td>51.6637</td>
<td>56.3618</td>
</tr>
<tr>
<td></td>
<td>Q</td>
<td>0.9874</td>
<td>0.9945</td>
</tr>
<tr>
<td>Elaine</td>
<td>Capacity (bit)</td>
<td>31081</td>
<td>753695</td>
</tr>
<tr>
<td></td>
<td>PSNR (dB)</td>
<td>51.9693</td>
<td>55.8437</td>
</tr>
<tr>
<td></td>
<td>Q</td>
<td>0.9907</td>
<td>0.9953</td>
</tr>
<tr>
<td>Baboon</td>
<td>Capacity (bit)</td>
<td>96340</td>
<td>102574</td>
</tr>
<tr>
<td></td>
<td>PSNR (dB)</td>
<td>47.0073</td>
<td>51.0359</td>
</tr>
<tr>
<td></td>
<td>Q</td>
<td>0.9673</td>
<td>0.9857</td>
</tr>
</tbody>
</table>

As shown figure 4 and figure 5, histogram and stego image of Baboon are displayed with LSB-3, Yu’s and proposed method. It is seen that proposed technique is the best in three methods.

Figure 4. Comparison of histogram of Baboon.

Figure 5. Comparison of stego image of Baboon.

Figure 6. Stego image Baboon and retrieved image

It shows the stego image, relevant histogram and retrieved image with the proposed method in figure 6. Security is satisfied in proposed method. Three keys are used to embed and extract secret message.

This algorithm is considered a high visual effect due to the low modification on the host data that makes the stego image has a very good quality. The visual quality is measured by the PSNR and Q.

6. Conclusions

In this paper, the secret data is embedded in the edge of cover image only. That’s because the human visual system (HVS) is more sensitive to slight changes in smooth areas than edge regions. At the same time, Hamming Code is applied to protect secret data. Three keys are applied to improve security. The larger edge pixel value, the more bits it could be hidden. In order to alleviate the difference between stego image and cover image, 2k correction technique is applied. By experiment it is found that proposed approach is better than two other methods in capacity, Q and PSNR.

Acknowledgment

This research was financially supported by the project of the Natural Science Foundation of Fujian Province, China (Grant No. 2016J05153) and the National Natural Science Foundation of China (Grant No. 61473329).

References


Shuliang Sun graduated from Hangzhou Dianzi University, China, in 2003 and completed M.E. from Guangxi University in 2006, China. He received PhD from Tongji University in 2011, China. Currently, He is an Associate Professor at the School of Electronics and Information Engineering, Fuqing Branch of Fujian Normal University, China. His research includes Image Processing, Steganography and Pattern Recognition.