Data Hiding Technique Based on Dynamic LSB

Naziha M. Al-Aidroos
Marghny M. Mohammed
Mohammed A. Bamatraf
DATA HIDING TECHNIQUE BASED ON DYNAMIC LSB

Naziha M. AL- Aidroos*, Marghny H. Mohamed**, and Mohamed A. Bamatraf***

*Department of Mathematics, Faculty of Science, Assiut University, Egypt.
Naz_mj@aun.edu.eg

**Department of Computer Science, Faculty of Computers and Information, Assiut University, Egypt.
marghny@aun.edu.eg

***Department of Computer Science, Faculty of Science, Hadhramout University, Yemen.
mbamatrafl@yahoo.com

Abstract:

In this paper, we propose steganographic technique on images to provide higher capacity of secret information as well as imperceptibility of stego image for secret communication. To cover and recover hidden information, spatial domain approach of the image is used. The principle behind the proposed method is to increment the embedding capacity and with minimal effect in the stego image aiming high imperceptibility based on the simple LSB. The proposed technique uses variable number of LSB’s rather than fixed, for more efficiency the data hiding process selects set of edge pixels using pixel value difference (PVD) to minimize visual effects over the stego image. The experimental results show efficient performance of the proposed method compared to similar methods in the same domain, in terms of PSNR and the capacity in addition to visual effects. Efficiency of the model is evaluated from the viewpoint of both the insertion amount and the visual effects on the cover image (i.e. image quality). Moreover, the variable number of LSB’s improves the resistance against image steganalysis.

Keywords: Steganography, LSB Substitution, Pixel Value Difference, Data Hiding.

1. INTRODUCTION

In recent years, enormous research efforts have been invested in the development of digital image steganographic techniques. Image steganography is a secret communication technique used to transmit secret messages that have been embedded into an image. In image steganography, the original image and the embedded image are called the cover image and the stego image, respectively. The sender hides the secret message in a cover image that has no meaning, and then transmits the stego image to the receiver through a public channel. In the transmission process, the public channel may be intentionally monitored by some opponent who tries to prevent the message from being successfully sent and received. The opponent may randomly attack the stego image if he/she doubts the stego image carries any secret message because the appearance of the stego image shows obvious artifacts of hiding effect [4]. For this reason, an ideal steganography scheme, to keep the stego image from drawing attention from the opponent, should maintain an imperceptible stego image quality. That is to say, if there are more similarities between the cover image and the stego image, it will be harder for an attacker to find out that the stego image has important secret data hidden inside it [11]. This way, the secret data is more likely to travel from the sender to the receiver safe and sound.

For data hiding, many steganographic algorithms have been proposed to hide secret information in digital images. The most common method is the LSB (least significant bit) method, which utilizes some least bits of pixels in the cover image to embed the hidden data. The advantages of the LSB method are ease of computation and a large payload of data that can be embedded in the cover image with high visual quality.

Recently, some steganographic techniques have been reported which modify the pixels to embed data, such as, Wu et al. [9] proposed the pixel value differencing (PVD) method of steganography which can hide large amount of data by modifying the difference value set between pairs of adjacent pixels. Using this technique, more data can be inserted into areas where differences in the adjacent pixel values is large as pixels in these areas can tolerate more changes and this leads to good imperceptibility and a high embedding rate. The proposed method uses the difference value between pixels adjacent to the target pixel. The difference value is used to decide which pixels belong to an edge area and which are not, to embed the secret data bits within the spatial domain of these pixels by comparing the secret data bits with the similar corresponding pixel bits of the cover image pixels. We experiment on various standard images to evaluate the efficiency of the proposed method, and as a result, our method is able to produce stego image more similar to the original image. The rest of the paper is organized as follows. In section 2, LSB steganography technique, and edge detection are reviewed. Our steganographic algorithm for gray level images is presented in section 3. Experimental results
are illustrated and discussed in section 4. Finally, section 5 is our conclusion.

2. BACKGROUND

2.1. LSB STEGANOGRAPHY

Many steganographic methods embed a large amount of the secret information in the first k LSB’s of the cover image pixels. Because of the imperfect sensibility of the human visual system, the existence of the embedded secret information can be imperceptible.

A digital image I can be represented by a two dimensional. Each image is composed of finite elements each of which has a definite location and amplitude. These elements are referred as image pixels, \( I = \{ P_i, ..., P_N \} \), every pixel in the grey scale image consists of 8 bits: \( |P_i| = 8 \), \( P_i = \{ b_{i1}, ..., b_{i8} \} \) where: \( b_{ij} \in \{0,1\} \).

LSB with k embedding factor \( 1 \leq k \leq 8 \) for every \( P_i \) targeted for embedding data bits replaces the set of bit: \( P_i = t_{i1}, ..., t_{ir} \), keeping the rest of bits \( f_{i1}, f_{i2}, ..., f_{is} \) without effect.

The generated set of pixels \( \{ P'_1, ..., P'_N \} \) represents as the stego image \( I' \), where: \( P'_i = [b'_{i1}, ..., b'_{is}] \), \( b'_{ij} \in \{0,1\} \).

The following example describe how the LSB embedding is happen, suppose we have the following pixels, \( P_i = [11001011], P_j = [00011010], P_k = [01001100] \), and the bits want to embed it in the LSBs positions on them are, \( M = [010] \), the resulted pixels after embedding are, \( P_i = [111001011], P_j = [000110100], P_k = [010011000] \). Altering the bits in LSB position of the image pixels does not change the quality of it to human perception.

2.2. EDGE DETECTION

The edge detection process serves to simplify the analysis of images by drastically reducing the amount of data to be processed, while at the same time preserving useful structural information about object boundaries [1].

An edge is characterized by significant dissimilarity in gray levels being used to indicate the boundary between two regions in an image fragment. Edge detection is a significant area of the image processing and machine vision due to the fact that edges are considered to be the important features for analyzing the most essential information contained in images [2].

The LSB-based methods, directly embed the secret data into the spatial domain in an unreasonable way without taking into consideration the difference in hiding capacity between edge and smooth areas. In general, the alteration tolerance of an edge area is higher than that of a smooth area. That is to say, an edge area can conceal more secret data than a smooth area. While human perception is less sensitive to subtle changes in edge areas of a pixel, it is more sensitive to change in the smooth areas. In our proposed method the cover image is divided into an edge and smooth regions to benefit of this property. Several methods exploiting these human visual characteristics have been studied recently, such as [2], [3], [5], [6], [7], [8], [9], [10], [12].

3. THE PROPOSED SCHEME

Any image \( I \) consist of set of pixels:
\[
I = \{ P_i, ..., P_N \},
\]
\( |P_i| = 8 \) bits, \( P_i = \{ b_{i1}, ..., b_{i8} \} \) where: \( b_{ij} \in \{0,1\} \).

The image size is computed as:
\[
N = W \times H
\]
Where \( W, H \) is the image width and height respectively. Suppose \( M \) is the secret data bits, with length \( n \),
\[
M = \{ m_1, m_2, ..., m_n \}, \text{where } m_i \in \{0,1\}
\]
(3)
The maximum hiding capacity \( h \) in the image \( I \) in terms of bits is:
\[
1 \leq h \leq (N \times 8)
\]
(4)
Like other data hiding schemes, the proposed scheme consists of two procedures, the embedding procedure and the extracting procedure. In this section the procedure steps are described in detail.

3.1. DATA EMBEDDING PROCEDURE

The proposed method takes the dependency advantage of pixels on its surrounding neighbors. The correlation between a pixel and its neighbors decides whether it is located in smooth area or in an edge area. The data embedding algorithm’s steps are as follows.

**Step 1:** Obtain the edge image from the grayscale image \( I \), by dividing the image into overlapping blocks, each block consists of 4 neighboring pixels (Figure 1).

![Figure 1: A Target Pixel and Three Neighboring Pixels.](image)

Where the target pixel \( P_{(i,j)} \) with gray value \( g_{(i,j)} \); let \( g_{(i,j+1)} \), \( g_{(i+1,j)} \), \( g_{(i+1,j+1)} \) and \( g_{(i+1,j+1)} \) be the gray value of the neighboring pixels, right pixel as \( P_{(i,j+1)} \), down-right pixel as \( P_{(i+1,j+1)} \) and down pixel as \( P_{(i+1,j)} \) respectively.

**Step 2:** Calculate the difference value \( d \) of each block to indicate the smooth and edge regions, then select the maximum difference among \( d \) as following:
\[
d_1 = |g_{(i,j+1)} - g_{(i,j)}|, \\
d_2 = |g_{(i+1,j)} - g_{(i,j)}|, \\
d_3 = |g_{(i+1,j+1)} - g_{(i,j)}| \\
\text{d} = \text{max}(d_1,d_2,d_3)
\]
(5)

**Step 3:** Using Equation (5), we decide whether the target pixel is included in an edge area, if \( d \) value is more or equal to a certain threshold, otherwise it included in a smooth area, after this step we obtain two sets of pixels, the first called edge pixels denoted as \( (E) \), and the other called smooth pixels denoted as \( (S) \),
\[
E = \{E_1, ..., E_g\}
\]

**Figure 1:** A Target Pixel and Three Neighboring Pixels.
Where $e$, $s$ is the edge pixels size, and smooth pixels size respectively, where $N = e + s$.

**Step 4:** For embed data bits stream $M$ of size $n$, initially convert the pixels in the edge and smooth areas of the cover image to the binary system. At start, we fixed the embedding into first 2 positions of the smooth pixel which are least significant bits, as following:

\[ S = \{S_1, \ldots, S_j\} \]

(6)

\[
\forall \text{Pixel } S_i \in \text{Smooth Area } (S) \\
S_{i1} = M_k \\
\text{increment } k \text{ by 1} \\
S_{i2} = M_k \\
\text{increment } k \text{ by 1} \\
(7)
\]

Where $i$ is the smooth pixel index, $k$ is the secret data bits index. Then the embedding process is completed as shown in the following steps:

\[
\begin{align*}
    j &= 3 \\
    \text{if } S_{ij} &= M_k \\
    \text{embed in } S_{ij} \\
    \text{increment } k, j \text{ by 1} \\
    \text{Else, save } j \text{ in 3LSB of } E_i \text{ and stop}
\end{align*}
\]

(8)

Where $j$ is the smooth pixel bits index, the metadata about last embedding in pixel $S_i$ is saved in the 3LSB’s of $E_i$.

**Step 5:** After embedding all $M$, the stego image $I'$ is produced, then send it to the other side to extract the secret information.

The block diagram of the embedding process steps is shown in Figure 2.

**3.2. DATA EXTRACTING PROCEDURE**

To recover the original secret data at the receiving side, the original image $I$ must be known to determine the original edge ($E$) and smooth ($S$) areas before embedding, and then the following steps are done.

**Step 1:** Divide the original image $I$ as the same way described in the previous section into blocks with 4 neighboring pixels.

**Step 2:** Find an edge ($E$) and smooth ($S$) areas by calculating the difference values of the pixels, by the same way mentioned in section 3.1.

**Step 3:** Using the position of the original smooth and edge areas we can determine which pixels belong to the edge area ($E'$), and which are belong to the smooth area ($S'$) in the stego image ($I'$), easily.

**Step 4:** Extract the metadata defines the position of the last embedded bits in the smooth pixels $S'$ from the 3LSB of the edge pixels $E'$.

\[
\begin{align*}
    &\text{for } k = 1 \text{ to } n \\
    &\text{set metadata } = \text{3lsb’s of } E_i' \\
    &\text{for } j = 1 \text{ to metadata} \\
    &\text{increment } k \text{ by 1} \\
    &\text{go to next pixel in } S'
\end{align*}
\]

(9)

Where $M'$ is the extracted data bits.

**Step 5:** At this stage, the retrieving algorithm finishes and the embedded data has been retrieved completely. This step is described in the block diagram in Figure 3.
4. EXPERIMENTAL RESULTS

The experimental results presented in this section demonstrate the performance of our proposed scheme. To conduct our experiments, we used four 128×128 standard grayscale images, “Baboon”, “Lena”, “Pepper” and “Cameraman”. These images are shown in Figure 4.

A series of pseudo random binary numbers are used as the secret data to be embedded into the cover images. The peak signal-to-noise ratio (PSNR) was utilized to evaluate the stego image quality. The PSNR formula is defined as:

\[ \text{PSNR} = 10 \times \log_{10} \frac{10^{\text{MSE}}}{\text{dB}} \]  \hspace{1cm} (10)

Where \( \text{MSE} \) is the mean square error between the cover and stego images. For a cover image whose width and height are \( W \) and \( H \), \( \text{MSE} \) is defined as:

\[ \text{MSE} = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} (I_{ij} - \hat{I}_{ij})^2 \]  \hspace{1cm} (11)

Where \( I_{ij} \) and \( \hat{I}_{ij} \) are the pixel values of the cover and stego images, respectively. A higher PSNR value indicates that the quality of the stego image is better and that it is more similar to the cover image, this will prevent the detection visual attack by human eye.

Also as a performance measurement for embedding capacity, the average number of bits embedded into each pixel is calculated as:

\[ \text{Capacity} = \frac{\text{Total number of bits embedded into image}}{\text{Total number of pixels in image}} \]  \hspace{1cm} (12)

Table 1 shows the quality of the stego image generated by our proposed scheme. The stego images which are generated have a higher quality (greater than 37 dB) under the human visual system (HVS).

The average of PSNR, and capacity is calculated for 6 runs with different secret data bits length, the results of the proposed method for the four grayscale images are shown in Table 2.

The mentioned metrics could clearly prove the applicability of the proposed model. The selected metrics produced a set of results to be discussed and justified based on behavior of the algorithm.

The proposed scheme has several advantages. Firstly, in the smooth area the embedding is done based on similarity between the data secret bits and the smooth pixels bits, except the first two least significant bits, this means, the proposed method does not cause any change in the pixel bits of smooth area almost, the embedding process in the 2 LSB bits can be considered as the worst case of the model capacity, in case of no match is met in the remaining part of the pixel. At the same time in case of further match(s) is/are found capacity will be improved without causing any effect in image quality as the similarity between bits is the embedding criteria for the remaining bits. otherwise, in the edge area we just embed the metadata about last position we embed in the corresponding smooth pixels, only in the 3LSB’s of it, this indicates that our method does not change the image statistics to a noticeable extent. Secondly, it requires much less computational overhead as compared to other steganographic techniques which transform image in the frequency domain. Finally, mapping capacity is variable and can be decided depending on the number of pixels in the smooth and edge area, and the amount of data to be mapped. Also, it depends on how much the secret data is similar to the image pixels bits, searching for the higher similarity rather that blindly embedding the data. This minimizes the change in the same image pixels.

5. CONCLUSION

In this paper, a new method to embed secret data into the cover image by a dynamic LSB substitution method based on divide the cover image into edge and smooth areas, using the pixel-value differencing to improve the image quality of the stego image and increase the embedding capacity of the cover image. The main idea here is to utilize the similarity in the smooth area, unlike simple techniques that use the edge area. Because the feature that the human eye can’t detect the changes on edge area easily, we embed only the 3LSB’s on its pixels to save the metadata about the last occurrence of the embedded bits in the corresponding pixels. This advantages provided by the proposed scheme benefit in generating a higher quality stego images under the HVS more similar to the original image before embedding. Experimental results confirm that the proposed scheme is successful in obtaining a stego image of satisfactory quality. Moreover, it can resist steganalysis systems which are based on statistical analysis.
Table 1: The Stego Image Quality Generated by Our Scheme.

<table>
<thead>
<tr>
<th>Greyscale Image</th>
<th>Edge Image</th>
<th>The Number of Edge Pixels</th>
</tr>
</thead>
<tbody>
<tr>
<td>7097 Pixel</td>
<td>7401 Pixel</td>
<td>2803 Pixel</td>
</tr>
<tr>
<td>8676 Pixel</td>
<td>2803 Pixel</td>
<td>7401 Pixel</td>
</tr>
<tr>
<td>7097 Pixel</td>
<td>7401 Pixel</td>
<td>2803 Pixel</td>
</tr>
</tbody>
</table>

Table 2: Experimental Results for the Proposed Method.

<table>
<thead>
<tr>
<th>Cover Images</th>
<th>Avg. Capacity (Bits/pixel)</th>
<th>Avg. PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baboon</td>
<td>1.56</td>
<td>41.74</td>
</tr>
<tr>
<td>Lena</td>
<td>1.47</td>
<td>41.10</td>
</tr>
<tr>
<td>Pepper</td>
<td>1.45</td>
<td>41.40</td>
</tr>
<tr>
<td>Cameraman</td>
<td>1.54</td>
<td>39.00</td>
</tr>
</tbody>
</table>

REFERENCES:


