Myanmar Character Hiding into BMP Image Based on Hexadecimal-Digits Representation

Aye Chan Ko, Nay Aung Aung
University of Computer Studies, Mandalay
ayechanko.ucsm@gmail.com, nayaungaung.ucsm@gmail.com

Abstract

Steganography is the approach for hiding any secret message in a variety of multimedia carriers like text, images, audio or video files. Whenever hiding a secret message, it is very important to make it invisible. This paper proposes a technique to hide Myanmar text message into BMP image by representing Myanmar Unicode hexadecimal value of each character in the secret message. It also represents each hexadecimal pixel value in the cover image as a set of separated single hexadecimal-digit. The technique creates a matching list from the hexadecimal-digits of the characters in the secret message with the hexadecimal-digits of the pixels in the cover image. The technique compresses the created matching list to be as small as possible to embed it in the unused file space at the end of the cover image file. The results show that this technique provides security against visual attack because it does not make any changes in the pixel of the cover image.

Keywords: steganography, bmp image, Unicode, embedding, compression, security

1. Introduction

The term steganography means the science of hidden communication. The way in which steganography differs from another secure data communication technique called cryptography which is, the visibility of the data exchange [5]. In cryptography, even though the actual data transaction may not be known to a third person, he may get a doubt that some abnormal or suspicious communication is taking place. But, in case of steganography, the hidden communication will never come to the notice of the eavesdropper. This is because the carrier signal we are using to hide the secret data is going to be innocent [6].

Steganography, copyright protection for digital media and data embedding are the data hiding techniques [8]. Steganography is a method of hiding secret information using cover images. In data embedding systems the receiver will know about the hidden message and the task is to decode the message efficiently. The main aspect of steganography is to achieve high capacity, security and robustness.

The various steganographic techniques are: substitution technique, transform domain technique, spread spectrum technique, statistical technique and distortion technique [2].

BMP is a graphics format used commonly as a simple graphics file format on Microsoft Windows platform. Sometimes it is called DIB which stands for device-independent bitmap. The bitmap is a type of memory organization or image file format used to store digital images. The term bitmap comes from the computer programming terminology which means just a map of bits, a spatially mapped array of bits.

The hexadecimal digit can represent more efficient than other numerical digit representation in Unicode character representation. If we use two hexadecimal digit, we can be represent Unicode character range from 00 to FF (256 characters). In decimal representation, we can represent only 100 Unicode character.

In this paper, image steganography is used to hide information by performing a proposed kind of encryption. The Steganography technique is the perfect supplement for encryption that allows a user to hide a large amount of information within an image. Thus, if the sensitive information will be transmitted over unsecured channel such as the internet, steganography technique can be used to provide an additional protection on a secret message.

A good technique of image steganography aims at three aspects. First one is capacity (the maximum data that can be stored inside cover image). Second one is the imperceptibility (the visual quality of stego-image after data hiding) and the last is robustness (security against attacks) [3].

2. Related Works

In digital image steganography, Least Significant Bit (LSB) method is mostly used in hiding information inside image. The 8th bit (right most bit) is substituted by one bit of every bit of the secret information. Although the LSB method usually does not increase the file size, the file can become noticeably distorted depending on the size of the secret information [4].

P. Mohan Kumar et.al. [6] evaluated many thousands of natural images using different kinds of steganalytic algorithms. They showed that both visual quality and security of their stego-images improve significantly compared to typical LSB-based approaches and their edge adaptive versions.

H S Majunatha Reddy et.al. [2] proposed high capacity and security stegano-graphy using discrete wavelet transform algorithm. It is observed that the capacity and security are increased with acceptable PSNR compared to the existing algorithms.

Mohammed Abbas Fadhil Al-Husainy et.al. [3] proposed a good point of steganography based on decimal digits. In this steganography technique, there is no need to calculate the Peak Signal to Noise Ratio (P SNR) because there is no distortion in the pixels values of the stego-image. While the stego-image in all the steganography technique that is using the Least Significant Bit (LSB) to hide the secret message suffer from having some distortion that is appearing in the pixels values.

3. Image Steganography

An image steganography technique is presented in this section. This technique uses a bitmap images to hide a secret Myanmar message by employing a new way to encoding the secret message and hide it in the stego-image. Before going deep in the details of this technique, simple definitions that are adopt, in this technique, for the secret message and stego-image will be given below:

3.1. Secret Message

A secret message in this proposed image steganography technique is a Myanmar message which contains 33 consonants, 21 dependent and independent vowels, 10 digits, various signs and punctuation that are useful in writing any message to give the reader a good understanding of the message. As it is known, each character has a hexadecimal value that is representing a value of its Unicode code. Table 1 shows the Unicode code table of Myanmar characters [9].

3.2. Myanmar Script

The Myanmar script is used to write Burmese, the majority language of Myanmar (formerly called Burma). Variations and extensions of the script are used to write other languages of the region, such as Shan and Mon, as well as Pali and Sanskrit. The Myanmar script was formerly known as the Burmese script, but the term "Myanmar" is now preferred. The Myanmar writing system derives from a Brahmi-related script borrowed from South India in about the eighth century for the Mon language. The first inscription in the Myanmar script dates from the eleventh century and uses an alphabet almost identical to that of the Mon inscriptions. Aside from rounding of the originally square characters, this script has remained largely unchanged to the present. It is said that the rounder forms were developed to permit writing on palm leaves without tearing the writing surface of the leaf. Because of its Brahmi origins, the Myanmar script shares the structural features of its Indic relatives: consonant symbols include an inherent "a" vowel; various signs are attached to a consonant to indicate a different vowel; ligatures and conjuncts are used to indicate consonant clusters; and the overall writing direction is left to right.

3.3. Cover Image (BMP)

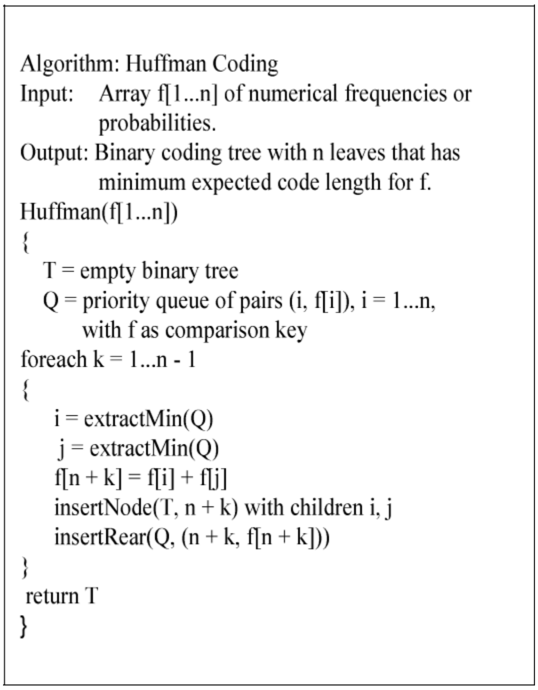
For the purpose of testing, a type of cover image that is candidate to be used in this work is a bitmap images (BMP). In general, each bitmap file contains a bitmap-file header, a bitmap-information header, a color table, and an array of bytes that defines the bitmap bits. The BMP Bitmaps are defined as a regular rectangular mesh of cells called pixels. Each pixel contains a color value. Bitmaps are characterized by only two parameters: the number of pixels, and the information content (color depth) per pixel, and they are the most commonly used type to represent images on the computer.BMP is the native bitmap format of Windows. BMP is a general format that stores images in different color depths without compression [1].

While the pixels of each image are representing as a two dimensional array, the proposed technique looks to the pixels of the image as a one dimensional array list of bytes, which have values between (00…FF), by reading the bytes of the two dimensional image row by row and stores them as a one dimensional array list.

Obviously, the operating system of any computer system stores files in the digital storage as a set of unified size of blocks (i.e., Kilobytes, Megabytes, and Gigabytes). This means that, for example, when we have a file of size (3920 bytes). This file will be stored in the digital storage as 4 Kilobytes (where: 1 Kilobyte=1024 bytes), such an operating system strategy remains 176 bytes unused at the end of this file. These unused bytes at the end of the (BMP) image file will be used by the proposed steganography technique to store the encoding information of the secret message within the cover image [3].

3.4. Huffman Coding Compression algorithm

Huffman's algorithm constructs a binary coding tree in a greedy fashion, starting with the leaves and repeatedly merging the two nodes with the smallest probabilities. A priority queue is used as the main data structure to store the nodes [7]. The Huffman Coding Compression algorithm is as shown in Figure 1.



```
Algorithm: Huffman Coding
Input:   Array f[1...n] of numerical frequencies or
         probabilities.
Output: Binary coding tree with n leaves that has
         minimum expected code length for f.
Huffman(f[1...n])
{
   T = empty binary tree
   Q = priority queue of pairs (i, f[i]), i = 1...n,
        with f as comparison key
foreach k = 1...n - 1
{
    i = extractMin(Q)
    j = extractMin(Q)
    f[n + k] = f[i] + f[j]
    insertNode(T, n + k) with children i, j
    insertRear(Q, (n + k, f[n + k]))
}
 return T
}
```

Figure 1. Huffman Coding Algorithm

4. Proposed System Design

The embedding process and extracting process are as shown in Figure 2 and 3.

4.1. Preprocessing

First of all, the technique changes the Myanmar Unicode value of the alphabetic characters in Table 1 to be represented in two hexadecimal digits only by each Unicode value. The new sequence of the Unicode values of the alphabetic character becomes (00…FF). The modified Table1will be shown as Table 2. After that, the technique creates a list M. Each element in this list is representing one hexadecimal digit of the Unicode of the characters in the secret message. The length of the list M (i.e., number of elements) is half of the length of the secret message (i.e., number of characters in it). For example:

Secret Message:

---- M: 1A 2F 14 32 48 2F 14 1E 2A 1E 0A 34 44

In the same manner, the technique generates a list D from the one dimensional array list of bytes of the stego-image from cover image, such that each element in this list is represented by hexadecimal values. For example:

One dimensional array list of image D: 10A2B7F00A189C432D91482F7012CD141E…

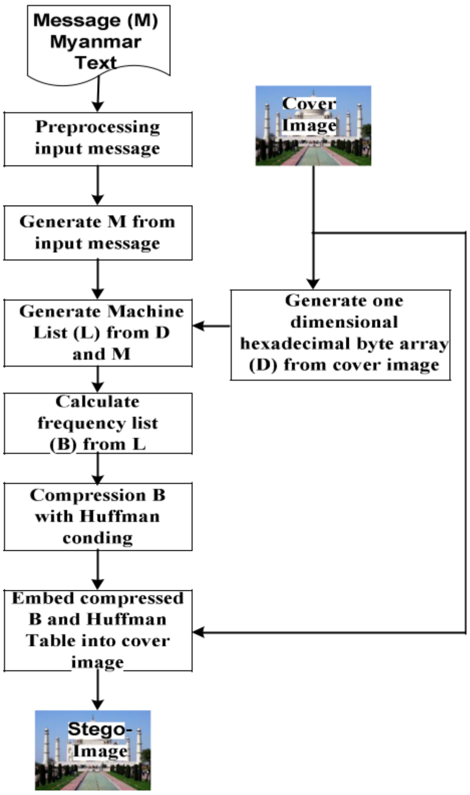

Message (M) Myanmar Text
Preprocessing input message
Generate M from input message
Generate Machine List (L) from D and M
Calculate frequency list (B) from L
Compression B with Huffman conding
Embed compressed B and Huffman Table into cover image
Stego-Image
Cover Image
Generate one dimensional hexadecimal byte array (D) from cover image

Figure 2. Embedding process

4.2. Generating Matching List (L)

The main operation of the proposed technique is creating the match list L by finding the match between each element in the list M with the elements in the list D, where each element in the list L represents either 0 or 1, (where 0: false/not match and1: true/match).To clarify this operation, consider the list M and D from the above two examples:

M:1A2F1432482F141E2A1E0A3444

D:10A2B7F00A189C432D91482F7012CD141E……

L:1011001000100011100011110010000111…..

After create the matching list L, re-represent the list L as a new frequency list B, such that each element in B represents the number of continuous values of 0s or 1s in the list L. For example, for the above created list L, the list B will be as follow:

B: 112213133342143……

After that, if the left most element of L is 1, append the 1 to the left most element of B. If it is 0, append the 0.

B:1112213133342143…….

4.3. Compression B List Using Huffman Coding

After the list B had generated, the proposed technique try to minimize the size (in byte) that is required to store this list in the unused bytes at the end of the (.bmp) stego-image file. One of the most well-known compression methods is Huffman coding method; this is used to compress the B list.

4.4. Embedding Process

When the proposed technique complete the above three steps, all the necessary information about the secret message will be embed in the unused bytes at the end of the (.bmp) cover image file. This information includes: the compressed B list and the Huffman coding table.

4.5. Extracting Process

When the receiver get the file of the stego-image, it is easy to read all the information about the embedded secret message from the unused bytes space at the end of the (.bmp) stego-image file. Then decompress the B list by using the Huffman coding table and then convert it to the list L. After that, the receiver extracts the list M from the two lists D and L by finding the match positions in the list D. At the end of the extracting process, use the list of frequencies to reconstruct the original secret message from the list M. The extracting original secret message is as follow:

Step1: extract the compressed B list from stego- image

Step2: decompress B list

B:1112213133342143…….

Step3: generate matching list L from B list. If the left most element is 1, the matching list L starts with 1. If not, L starts with 0.

L:1011001000100011100011110010000111…..

Step4: generate one dimension hexadecimal Value byte array from received stego-image.

D:10A2B7F00A189C432D91482F7012CD141E….

Step5: generate M list from D and L.

D:10A2B7F00A189C432D91482F7012CD141E….

L:1011001000100011100011110010000111…..

M:1A2F1432482F141E2A1E0A3444

Step6: reconstruct original secret Myanmar message by using Table 2 and M.

```
1A 2F 14 32 48 2F 14 1E 2A 1E 0A 34 44
 -         -         -        -
```

Original Message:

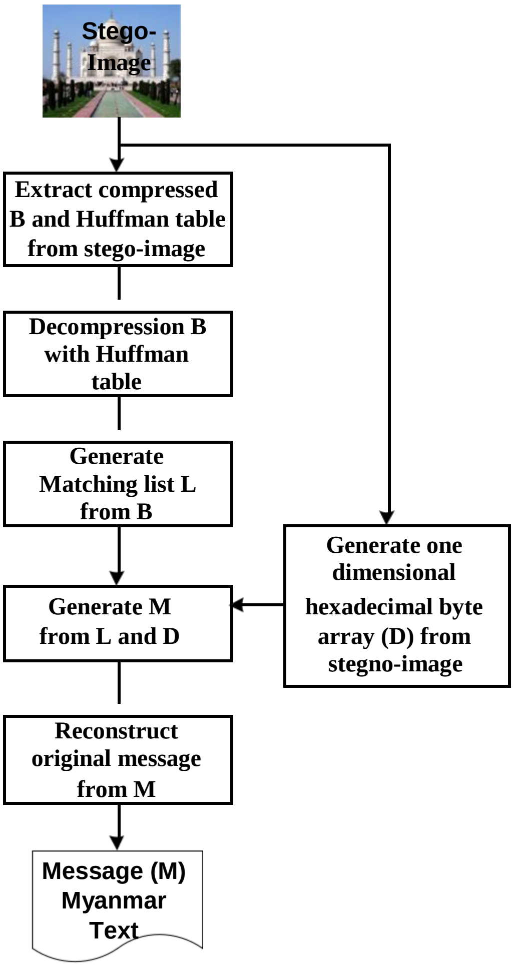
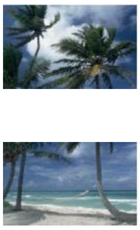

Stego-Image
Extract compressed B and Huffman table from stego-image
Decompression B with Huffman table
Generate Matching list L from B
Generate M from L and D
Generate one dimensional hexadecimal byte array (D) from stegno-image
Reconstruct original message from M
Message (M) Myanmar Text

Figure 3. Extracting process

5. Experimental Results

In the proposed system, some experiments are tested by using different cover bmp images of different sizes to hide randomly secret Myanmar messages of different length as shown in Table 3 .and 4.


Beach (360*540*24)

Bridge (600*394*24)

Taj-Mahal (600*494*3)

Lighthouse (1920*1200*24)

Figure 4. Cover images(BMP)

Table 3. Experimental results

Cover-image	Beach	Bridge	Taj-Mahal	Lighthouse
Image Size(byte)	583200	709200	889200	6912000
Number of Unused byte	426	378	602	970
Size on Disk (byte)	583680	709632	889856	6913024
Length of Secret Message (character)	456	400	762	1195
Size of Embedded Information (byte)	399	352	568	944

The image size and the size of file allocated on Disk can be calculated by the following equations:

Image Size = Width × Height × N/8 (1)
Header Size of BMP= 54 bytes (2)
Size on Disk = Image Size + Header Size + Number of Unused Bytes (3)

6. Conclusion

In this system, a propose technique to hide Myanmar text message inside images was presented. The main objective of this technique is to add more security to the message by applying some types of coding to the characters in the secret message and saving the pixels values of the cover image with no change by hiding all the secret information about the message in the unused bytes at the end of the cover image file. This can be done by using any Unicode code values that are representing in maximum two hexadecimal-digits (i.e., 00…FF).The proposed technique uses the Huffman coding method to compress the list B. The proposed technique does not make any changes in the pixels values of the cover image, and tries to hide all the secret information of the message in the unused space at the end of the cover image file. This is a good point in this steganography technique and there is no need to calculate the Peak Signal to Noise Ratio (PSNR) because there is no distortion appears in the pixels values of the cover image. Therefore the system is imperceptibility because the attackers don't believe that stego-image contains any secret information.

The future works include: increasing the capability of this technique by finding the best matched block of pixels in the cover image with the characters of the secret message, trying to provide the technique immunity about any rotation and resizing operations that might be done on the cover image, applying this technique on another types of stego-files like (text, audio, video).

References

[1] Anjali Anand and Dr. Himanshu Aggarwal , "BMP to JPEG – the Conversion Process", Department of Computer Engineering, Punjabi University, Patiala, An International Journal of Engineering Sciences ,ISSN: 2229-6913 , July 2011, Vol. 1.

[2] H S Majunatha Reddy and K B Raja, "High Capacity and Security Steganography Using DiscreteWavelet Transform", International Journal of Computer Science and Security (IJCSS), Dept. of Electronics and Communication Global Academy of Technology, Bangalore, India-560098.

[3] Mohammed Abbas Fadhil Al-Husainy, "A New Image Steganography Based on Decimal-Digits Representation", Department of multimedia systems, Al-Zaytoonah University of Jordan, Amman-Jordan, November 2011.

[4] Mohammad Ali BaniYounes, & AmanJantan, "A New Steganography Approach for Image Encryption Exchange by using the LSB insertion", IJCSNS International Journal of Computer Science and Network Security, 8(6), 247-254.,2008.

[5] N. Provos and P. Honeyman, "Hide and seek: an introduction to steganography", IEEE Sec. Priv. Mag., vol. 1, no. 3, pp. 32–44, 2003.

[6] P. Mohan Kumar and K. L. Shanmuganathan, "Developing a Secure Image Steganographic System Using TPVD Adaptive LSB Matching Revisited Algorithm for Maximizing the Embedding Rate", CSE Department, Jeppiaar Engineering College, Chennai, India, February, 2011.

[7] Prof. Sergio A. Alvarez," CS383, Algorithmsff Notes on Lossless Data Compression and Hu man Coding", Computer Science Department, Boston College

[8] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding",IBM Syst. J., vol. 35, no. 3–4, pp. 313–336, 1996.

[9] http://www.unicode.org/errata/ for an up-to-date list of errata.