FPGA IMPLEMENTATION OF LSB REPLACEMENT STEGANOGRAPHY USING DWT

M.Sathya[1], S.Chitra[2]
Assistant Professor, Prince Dr. K.Vasudevan College of Engineering and Technology

ABSTRACT

An enhancement of data protection system for secret communication using reserve room in encrypted images based on texture analysis with discrete wavelet is proposed here. The wavelet will decompose the image into four frequency sub bands namely LL, LH, HL and HH. These coefficients are then utilized in the encoder for removing the redundancies. The Selective embedding is utilized in this method to determine host signal samples suitable for data hiding. This approach uses the Least Significant Bits (LSB) insertion to hide data within encrypted image data. The binary representation of the hidden data is used to overwrite the LSB of each byte within the encrypted image randomly. This method proves to be more secure technique for secret data communication with high quality factor. The simulation results indicate that the framework can be successfully utilized in Image data hiding applications. The design utilizes the Spartan III EDK FPGA of Xilinx and LSB steganography algorithm to perform the steganography steps.

Index Terms: Steganalysis, adaptive steganography, selection channel, JPEG, detection, security.

1. INTRODUCTION

Maintaining the secrecy of digital information when communicated over the Internet is presently a challenge. Given the amount of cheap computation power available and certain known limitations of the encryption methods it is not too difficult to launch attacks on cipher-text. An ideal steganographic technique embeds message information into a carrier image with virtually imperceptible modification of the image. The objective of steganography is a method of embedding a additional information into the digital contents, that is undetectable to listeners. The idea behind the LSB algorithm is to insert the bits of the hidden message into the least significant bits of the pixels.

An example of such manipulations is insertion of secret information which is often referred to as information hiding. A successful insertion of a message into an image is more difficult using color images than that of grayscale images. A successful information hiding should result in the extraction of the hidden data from the image with high degree of data integrity. This project presents an information hiding technique that utilizes lifting schemes to effectively hide information in color images.

2. BASIC CONCEPTS

Steganography

Steganography means to hide secret information into innocent data. Digital images are ideal for hiding secret information. An image containing a secret message is called a cover image. First, the difference of the cover image and the stego image should be visually unnoticeable. The embedding itself should draw no extra attention to the stego image so that no hackers would try to extract the hidden message illegally. Second, the message hiding method should be reliable. It is impossible for someone to extract the hidden message if she/he does not have a special extracting method and a proper secret key. Third, the maximum length of the secret message that can be hidden should be as long as possible. “Steganography is the art of hiding information in ways that prevent the detection of hidden messages”.

FPGA Description

Field Programmable Gate Arrays Popularly known as FPGAs is an alternative for implementation of digital logic in systems.

Image Format

Digital images are representations of two dimensional images using a binary format. For the purpose of this research, Portable Network Graphics (PNG) images are used that are of the true color image type. Each pixel in a true color image has a section for the red channel, the green channel, and the blue channel. Additionally, pixels in the PNG format may specify a fourth channel called the alpha channel which stores the transparency of the pixel. Each channel contains the same number of bits (bit depth). A bit depth of 8 means that each channel of a pixel can contain a value in the range 0 to 255. The usual bit depths are 8 and 16 with 8 being the most common; however, other bit depths are possible. For this research, a bit depth of 8 is used for all images. For true color PNG images with a bit depth of 8, each pixel is stored using four bytes. There are three bytes that represent the color channels and one byte that represents the alpha channel. The red channel is stored in the second byte in bit positions 16-23. The green channel is stored in the third byte in bit positions 8-15 and the blue channel is stored in the fourth byte in bit positions 0-7. Each of these channels has a possible value between 0 and 255 with 0 being black and 255 being pure red, pure green or pure blue depending on the channel in which the value resides. All images in this research are single layer images meaning that no composite image is used or create.

Wavelet Transform

Wavelets are mathematical functions defined over a finite interval and having an average value of zero that transform data into different frequency components, representing each component with a resolution matched to its scale. The basic idea of the wavelet transform is to represent any arbitrary function as a superposition of a set of such wavelets or basis functions. These basis functions or baby wavelets are obtained from a single prototype wavelet called the mother wavelet, by dilations or contractions (scaling) and translations (shifts).

Discrete Wavelet Transform

Calculating wavelet coefficients at every possible scale is a fair amount of work, and it generates an awful lot of data. If the scales and positions are chosen based on powers of two, the so-called dyadic scales and positions, then calculating wavelet coefficients are efficient and just as accurate. This is obtained from discrete wavelet transform (DWT).

Image Embedding Process

Least Significant Bit Insertion (LSB); Characters in the ASCII code can be represented using 8 bits. The value of discrete coefficients can be manipulated slightly without being noticed by visual inspection after the image is reconstructed using the manipulated lifting coefficients. This research project is based on the premise that the bits of ASCII 6 characters can be included in lifting coefficients without resulting in a visible appearance. The idea behind the LSB algorithm is to insert the bits of the hidden message into the least significant bits of the pixels.

Process of Adding a Message: The process of adding a message to the pixels of an image is a multi-step process. In brief, an ASCII character stream is split into two-bit pairs, a lifting scheme is applied to an image, the two-bit pairs is inserted into the image in either the trends or details in the lifting domain, and then the inverse lifting process is applied to reconstruct the image. The channels of the image pixels are split into separate arrays upon initialization. There is one array for each color channel. Then it calculates the entropy and retrieves the text that has been entered by user.

Encoding the Message in the Image: Before manipulating the array of characters that is read in, a terminator is added to the end of the array. The terminator is three ‘*’ characters in a row. This terminator is used during the decode process to signal the end of the message input by the user in the encode process. The program then splits the ASCII character stream into 4 two-bit pairs per character. The two-bit pairs are created because the 2 LSBs of a pixel will be replaced with these two-bit pairs. Since all the characters in the English alphabet are within the lower 127 ASCII characters, only 4 two-bit pairs are needed to represent each character. These two-bit pairs are stored in an array for later manipulation. The lifting scheme is then applied to the image down to the level specified by the user. Since the image is split into three color channels, the discrete scheme must be applied three times, once for each color channel. The program automatically adjusts the encoding according to the discrete decomposition level.

Once the transformations are complete, the two-bit pairs of the ASCII characters are then hidden in the pixels of the processed image. The text offset value of the color channel specified by the user determines which bits are used to hide each subsequent two-bit pair of the ASCII character. Hiding the two-bit pair in the image is accomplished by overwriting the two selected bits of a pixel with the value of the two-bit pair. This is done by performing a bitwise AND operation with 0 and the two bits of the pixel, which effectively sets the two bits to 0. Then the two-bit pair to be hidden in this pixel is then combined with the pixel by a bitwise OR operator, effectively setting these pixel bits to the message bits.

Decoding a Message from the Image: Decoding a message that is inserted into an image requires fewer steps than to encode. The process flow starts out the same way as encoding with the user selecting the parameters that were used to encode the message. At this point the program splits the image into its color channels and applies the inverse discrete scheme to each channel to the level specified by the user. When the discrete transformation is completed, the program retrieves the message out of the pixels of the cover image.

3. EXPERIMENTAL RESULTS

Steganography technique is implemented such that a key is given at the transmitter side and the secret data can be obtained at the receiver by using the same key. The Discrete Wavelet Transform (DWT) will decompose the image into four frequency sub bands namely LL, LH, HL and HH. These coefficients are then utilized in the encoder for removing the redundancies. The LL portion of the decomposed image is chosen for hiding the data because this portion is similar to that of the input image. This approach uses the Least Significant Bits (LSB) insertion to hide data within encrypted image. LSB encoding is preferred to get a high clarity image. The binary representation of the hidden data is used to overwrite the LSB of each byte within the encrypted image randomly. Least significant bit replacement is effectively used for data hiding process. This method proves to be a more secure technique for secret data communication with high quality factor. At the receiver side Inverse Discrete Wavelet Transform (IDWT) and LSB decoding technique is used to get the secret data and the image separately. The simulation results indicate that the framework can be successfully utilized in Image data hiding applications.

3.1 Embedding Process

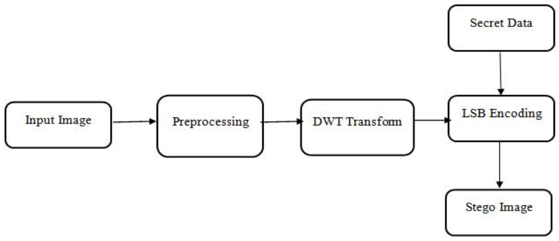

Fig.3.1 Embedding Process

3.1.1.Input Image

It represents the image in which the data has to be hidden. An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person. Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

Fig.3.2 Input Image

In wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo- photograph.

3.1.2 Preprocessing

In preprocessing, the image is resized to get a correct size and also gray conversion take place here.

Image resizing

Resize the image, this time specifying the desired size of the output image. Pass imresize a vector that contains the number of rows and columns in the output image. If the specified size does not produce the same aspect ratio as the input image, the output image will be distorted. If you specify one of the elements in the vector as NaN, imresize calculates the value for that dimension to preserve the aspect ratio of the image. To perform the resizing required for multi-resolution processing

Read image into the workspace.

```
[x,map]=imread('trees.tif');
```

Resize the image, specifying a scale factor. By default, imresize returns an optimized color map with the resized indexed image.

```
[y,newmap]=imresize(x,map,0.5);
```

Display the original image and the resized image.

```
Figure
Imshow(x,map);
Title('original image')
Figure
Imshow(y,newmap);
Title('resized image')
```

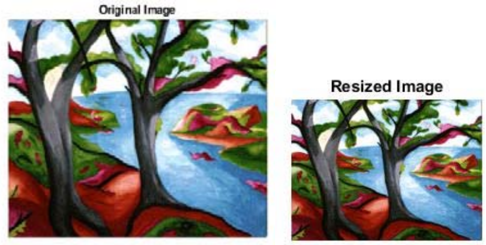

Fig.3.3 Image Resizing

Gray Conversion

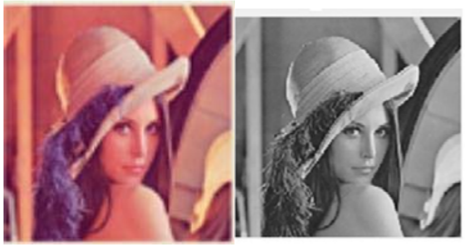

Fig.3.4 Gray Conversion

The color image consists of primary colors such as red, green and blue. All these are converted into gray scale with intensities from 0 to 255. This can be done by taking the average of the three colors. Since its an RGB image, so it means that you have to add r with g with b and then divide it by 3 to get the desired gray image.

Discrete Wavelet Transform

Calculating wavelet coefficients at every possible scale is a fair amount of work, and it generates an awful lot of data. If the scales and positions are chosen based on powers of two, the so-called dyadic scales and positions, then calculating wavelet coefficients are efficient and just as accurate. This is obtained from discrete wavelet transform (DWT).

Figure 3.5 shows the example for DWT process

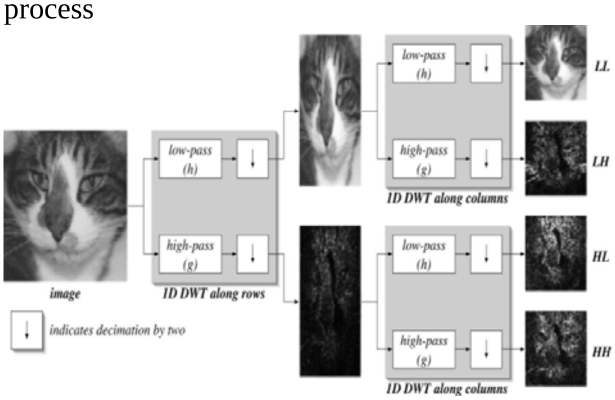

Fig.3.5 DWT Process

LSB ENCODING

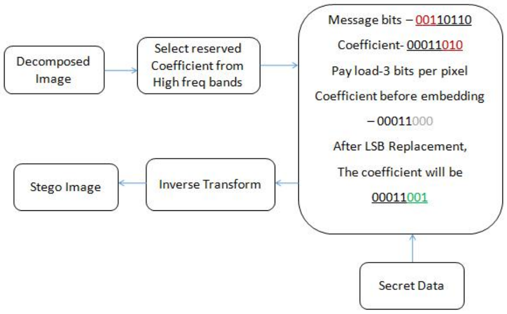

Fig.3.6 Algorithm Flow

Extraction Process

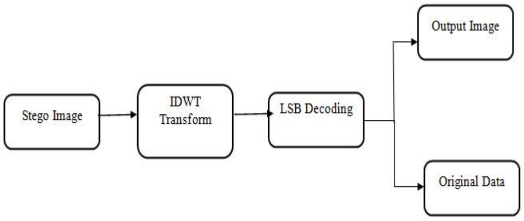

Fig.3.7 Extraction Process

Input Image

Input image is an image in which the data is going to be hidden. Here we are taking a 256 x 256 image. The input image is a color image such that a RGB image. Figure 3.8(a) shows the input image.

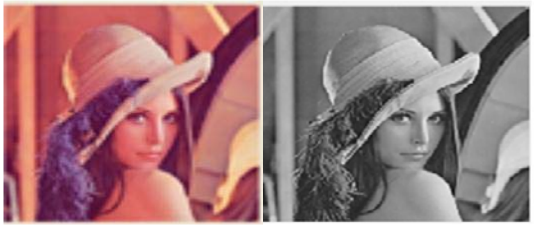

Fig.3.8 (a)Input Image (b) Gray Scale Image

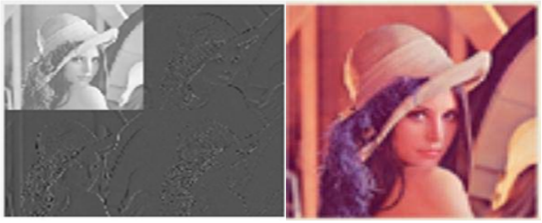

Fig.3.9 (a)Decomposed Image (b) Stego Image

4. CONCLUTIONS

In this a data hiding method by LSB substitution process is proposed. The image quality of the stego-image can be greatly improved with low extra computational complexity. Experimental result shows the effectiveness of the proposed method. In the proposed algorithm, the number of steps are very less. Thus, the computational complexity is reduced, so it is easy to be implementing in both grayscale and color image. Benefited from the effective optimization, a good balance between the security and the image quality is achieved.

REFERENCES

1. R. Cogranne, J. Fridrich,and V. Sedighi, “Content adaptive pentary steganography using the multivariate generalized Gaussian cover model,” in Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics2016(A. Alattar and N. D. Memon, eds.), vol. 9409, (San Francisco, CA), February 8–12, 2015.
2. T. Denemark, J. Fridrich, and V. Holub, , “Universal distortion design for steganography in an arbitrary domain,” EURASIP Journal on Information Security, Special Issue on Revised Selected Papers of the 1st ACM IH and MMS Workshop, vol. 2014:1, 2014.
3. J. Fridrich and V. Holub , “Designing steganographic distortion using directional filters,” in Fourth IEEE International Workshop on Information Forensics and Security, (Tenerife, Spain), December 2–5, 2012.
4. J.Fridrich and V. Holub, “Low complexity features for JPEG steganalysis using undecimated DCT,” IEEE Transactions on Information Forensics and Security, vol. 10, no. 2, pp. 219–228, 2015.
5. J. Fridrich and V. Holub, “Phase-aware projection model for steganalysis of JPEG images,” in Proceedings IS&T, ElectronicImaging, Media Watermarking, Security, and Forensics 2016(A. Alattar and N. D. Memon, eds.), vol. 9409, (San Francisco, CA), February 8–12, 2015.
6. L. Guo, J. Ni, and Y. Q. Shi, “Uniform embedding for efficient JPEG steganography,” IEEE Transactions on Information Forensics and Security, vol.9,no. 5, pp. 814–825, 2014.
7. L. Guo, J. Ni, and Y. Q. Shi, “An efficient JPEG steganographic scheme using uniform embedding,” in FourthIEEEInternational Workshop on Information Forensics and Security,(Tenerife, Spain), pp. 169–174, December 2–5, 2012.
8. J. Huang H. Li, W. Luo, and W. Tang, “Adaptive steganalysisagainst WOW embedding algorithm,” in 2nd ACM IH& MMSec. Workshop(A. Uhl, S. xxviii Katzenbeisser, R. Kwitt, and A. Piva, eds.), (Salzburg, Austria), pp. 91–96, June 11–13, 2014.
9. J. Huang , B. Li, X. Li and X. Li M. Wang, “A new cost function for spatial image steganography,” in Proceedings IEEE, International Conference on Image Processing, ICIP, (Paris, France), October 27–30, 2014.
10. F. Liu, X. Luo, X. Song, C. Yang and Y. Zhang, “Steganalysis of adaptive JPEG steganography using 2D Gabor filters,” in 3rd ACM IH& MMSec. Workshop (P. Comesana, J. Fridrich, and A. Alattar, eds.), (Portland, Oregon), June 17–19, 2015.