International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 04 Issue: 08 | Aug -2017 www.irjet.net p-ISSN: 2395-0072

IMAGE STEGANOGRAPHY METHOD USING ZERO ORDER HOLD ZOOMING AND REVERSIBLE DATA HIDING

[*1] Ms. Gayathri ., [*2] Mrs. Anita Madona N

*[*1] M.Phil Research Scholar, PG & Research Department of Computer Science & Information Technology Auxilium College , Vellore, Tamil Nadu, India*
*[*2]. Assistant Professor, PG & Research Department of Computer Science & Information Technology Auxilium College, Vellore, Tamil Nadu, India*

Abstract: *Steganography techniques are used to secure the secret message transmitted over an open communication channel such as the internet. It aims to embed secret data into a digital cover media, such as digital audio, image, video, etc., without being suspicious looking. New image steganography method that hides the secret message inside the cover image using zero order hold (ZOH) is considered. The major drawback of these methods is amount of additive noise that creeps in the image which directly affects the PSNR and the statistical properties of the image. Moreover the embedding algorithms are applicable mainly to lossless image-compression schemes like TIFF images. Steganography (which is the point of research) is a method in which the secret information is hidden inside a carrier fileA novel method by with a traditional RDH algorithm, and thus it is easy for the data hider to reversibly embed data in the encrypted image. Zero Order Hold (ZOH) and Least Significant Bit (LSB) are the technique for providing until cover image. In LSB-3, ZOH and traditional RDH algorithm is used for providing more security to embed into stego image. The PSNR values of the proposed method (ZOH) were higher than other methods, this means the method's PSNR (stego image quality) is improved. Reversible Data Hiding (RDH) technique is a secure way of transmitting data inside a cover media, so that data and cover file can be properly recovered by the receiver. The main goal of this method is to hide a secret message in the pixels of the cover image in such a way that the human eyes are not able to differentiate between the original and the stego-image.*

Keywords: ***Zero order Hold, Peak signal noise ratio, Reversible Data Hiding, Lease Significant Bit, stego images.***

I. INTRODUCTION

Over the last two decades, the rapid development of internet requires confidential information that needs to be protected from the unauthorized users. This is accomplished through data hiding method to hide secret messages into a cover file. Steganography techniques are used to secure the secret message transmitted over an open communication channel such as the internet. But message transmission over the internet is facing some problems [2]. So securing communication channel for transmitting data over the internet is needed, for these two schemes are used to protect secret messages from being stolen during transmission. The first scheme is a cryptography which is a well-known method in which the information is encrypted by using a key. Only the right person with a right key can decode and recover the original information successfully. The second one is a Steganography method in which the secret information is hidden inside a carrier file [2][3].

The ZOH techniques use the pixel gray levels and their color values directly for encoding the message bits. The ZOH techniques are the simplest schemes in terms of embedding and extraction complexity [3]. The major drawback of these methods is amount of additive noise that creeps in the image which directly affects the PSNR and the statistical properties of the image. Moreover the embedding algorithms are applicable mainly to lossless image-compression schemes like TIFF images. For lossy compression schemes like JPEG, some of the message bits get lost during the compression step [8].

The most common algorithm belonging to this class of techniques is the Least Significant Bit (LSB) replacement technique in which the least significant bit of the binary representation of the pixel gray levels is used to represent the message bit. This kind of embedding leads to an addition of a noise of 0:5p on average in the pixels of the image where p is the embedding rate in bits/pixel. It also leads to an asymmetry and a grouping in the pixel gray values (0,1); (2,3)... (254, 255) this asymmetry is exploited in the attacks developed for this technique.

The embedding is performed using a pair of pixels as a unit, where the LSB of the first pixel carries one bit of information, and data hiding carries another bit of information. It has been shown that embedding in this fashion reduces the embedding noise introduced in the cover signal [1][7]. In a multiple base number system has been employed for embedding data bits. While embedding, the human vision sensitivity has been taken care of. The

variance value for a block of pixels is used to compute the number base to be used for embedding.

II. EFFECTIVE WORK

Generally a Steganography system has a cover file that is used to cover the original message and the steganography algorithm to carry out the required object as shown in Fig.2.1 The result is a file called stego-file which has the message inside it, hidden[4]. This stego file is then sent to the receiver where the receiver retrieves the message by applying the de-steganography. The goal of modern steganography is to keep the message undetectable.

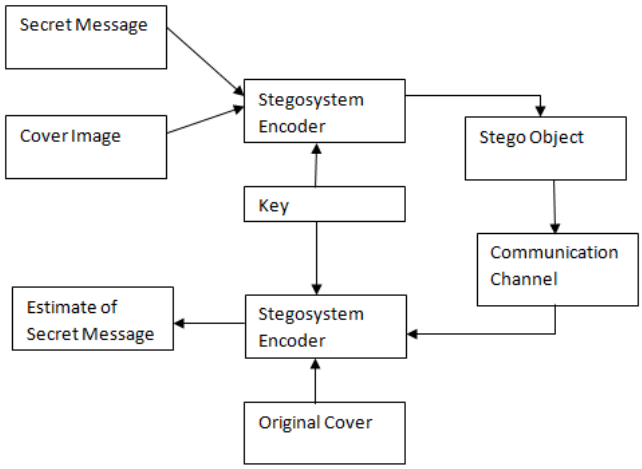

Fig : Steganography Model

Sending encrypted messages frequently will draw the attention of third parties, i.e. crackers and hackers, perhaps causing attempts to break and reveal the original messages. In a digital world, Steganography is introduced to hide the existence of the communication by concealing a secret message inside another unsuspicious message. Steganography is often being used together with cryptography and offers an acceptable amount of privacy and security over the communication channel. This paper presents an overview of text Steganography and a brief history of Steganography along with various existing text-based Steganography techniques L. Y. POR, B. Delina [1](2008). The proposed method hides the secret message inside the cover image by representing the secret message characters using Braille method of reading and writing for blind people that can save more than one-fourth of the required space for embedding. From the experimental results it is seen that the proposed method achieves higher visual quality as indicated by the high peak signal-to-noise ratio (PSNR) in spite of hiding a large number of secret bits in the image Abdelmgeid Amin Ali, Al – Hussien Seddik Saad (2013) [2]. overview of image Steganography, its uses and techniques. It also attempts to identify the requirements of a good Steganography algorithm and briefly reflects on which Steganography techniques are more suitable for which applications[3] Jagvinder Kaur, Sanjeev Kumar(2011). One of the methods used in security areas is Steganography. Steganography is the art and science of hiding information by embedding messages within cover media without attracting attention. The cover media can be text, image, video or audio files. Text Steganography is more difficult than others due to the difficulty in finding redundant information in text file. This paper presents a new idea for text Steganography by using Unicode standard characters, (which have the non-printing properties) to encode the letters of English language and embedding the secret message letter by letter into the cover-text. This method has high hiding capacity, it can hide (K+1) letters in a text with K characters and it does not make any apparent changes in the original text. So it satisfies perceptual transparency [4] Akbas E. Ali(2010).

III. PREVIOUS IMPLEMENTATIONS

The main terminologies used in steganography systems are: the cover file, secret message, stego file, embedding algorithm and extraction algorithm. The cover file is defined as the original file such as image, video, audio, text, or some other digital media used to embedding the secret message. The secret message is defined as the message you want to embed inside the cover file, it is called payload. Stego file is defined as the file after embedding the secret message in the cover file; it should have similar properties to that of the cover. The embedding algorithm is the method that used to embed the secret message in the cover image. The extraction algorithm is the method that retrieves the secret message from the stego image In the Steganography system, before the hiding process, the sender must select the carrier (i.e. image, video, audio or text) then select the secret message. The effective and appropriate Steganography algorithm must be selected that able to encode the message in more secure technique. Then the sender may send the Stego file by email or chatting, or by other techniques. After receiving the message by the receiver, the message can be decoded by using the extracting algorithm

Zero Order Hold Method

There are many methods for zooming image, Zero order hold is one of this. In zero order hold method, two adjacent elements are picked from the rows respectively and then the average value between two pixel (add them and divide the result by two then take the integer value) is calculated, and their result is placed in between those two elements. First, this row is done wisely and then the result is taken and do this column is don wisely as the same way. This is one of the simplest and easiest methods of hiding the data in images. In this method, the binary data form is

hidden into the LSBs of the carrier bytes or in pixels of image. The overall change to the image is so small that human eye would not be able to discover. In 24-bit images each 8-bit value refers to the red, green and blue color. But in 8–bit images each pixel is of 8-bits, so each pixel stores maximum 256 colors.

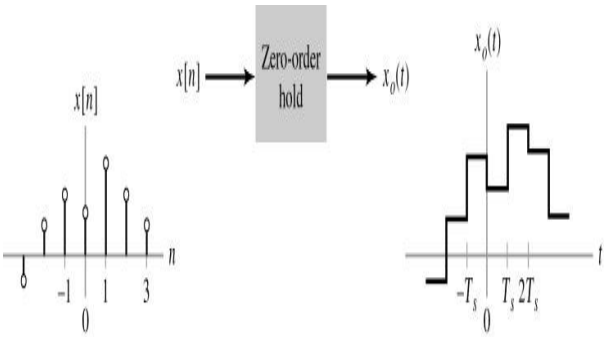

Fig : Zero order Embedding

ZOH Embedding Algorithm

Input: Cover Image C; Secret Message M.
Output: StegoImage S.
Steps:
1) Split C into 3 channels Red (R), Green (G), and Blue (B).
2) Convert image (B) to one column x.
3) Split M into characters.
4) Take m from M.
5) Convert m into binary bin.
6) Take pixel 1 from bin.
7) Calculate average of x (count) and x (count+1).
If end (average)! =end (bin) then x (count+1) = x (count+1) +2
8) Add 1 to count.
9) Repeat steps from 4 to 8 until the whole M has been embedded in C.
10) Merge the 3 channels R, G, y again to construct the StegoImage S.

Properties of z transformation

Linearity Let us consider summation of two discrete functions *f (k) and g (k)* such that

$$P\,x\,f(k) + qg(k)$$

such that p and q are constants, now on taking the Laplace transform we have by property of linearity:

$$Z[p\,x\,f(k) + q\,x\,g(K) = p\,x\,Z[f(k)] + q\,x\,Z[g(k)]$$

Change of Scale: let us consider a function f(k), on taking the z transform here have

$$Z[f(k)] = f(z)$$

The Zero-Order Hold block samples and holds its input for the specified sample period. The block accepts one input and generates one output, both of which can be scalar or vector. If the input is a vector, all elements of the vector are held for the same sample period. You specify the time between samples with the Sample time parameter. A setting of -1 means the Sample times inherited. A causal continuous-time signal x (t) under consideration is defined as

$$X(t) = \{x(t)\ for\ t \geq 0 \mid 0\ for\ t < 0\}$$

This block provides a mechanism for discrediting one or more signals in time, or resembling the signal at a different rate. If your model contains MultiMate transitions, you must add Zero-Order Hold blocks between the fast-to-slow transitions. The sample rate of the Zero-Order Hold must be set to that of the slower block. For slow-to-fast transitions, use the unit delay block. For more information about multi rate transitions, refer to the Simulink or the Real-Time Workshop documentation.

Message embedding Procedure

S(i,j) = C(i,j) - 1, if LSB(C(i,j)) = 1 and SM = 0
S(i,j) = C(i,j) + 1, if LSB(C(i,j)) = 0 and SM = 1
S(I,j) = C(i,j), if LSB(C(i,j)) = SM

Where LSB(C(i, j)) stands for the LSB of cover image C(i,j) and "SM" is the next message bit to be embedded. S(i,j) is the stego image. Here can use images to hide things if we replace the last bit of every color's byte with a bit from the message.
Message A-01000001
Image with 3 pixels
Pixel 1: 11111000 11001001 00000011
Pixel 2: 11111000 11001001 00000011
Pixel 3: 11111000 11001001 00000011

Algorithm Embedding Implementation

Begin
Input: Cover_Image, Secret_Message, Secret_Key;
Transfer Secret_Message into Text_File;
Zip Text_File;
Convert Zip_Text_File to Binary_Codes;
Convert Secret_Key into Binary_Codes;
Set BitsPerUnit to Zero;
Encode Message to Binary_Codes;
Add by 2 unit for bitsPerUnit;
Output: Stego_Image;
End

Algorithm for extracting Implementation

Begin
Input: Stego_Image, Secret_Key;
Compare Secret_Key;
Calculate BitsPerUnit;
Decode All_Binary_Codes;
Shift by 2 unit for bitsPerUnit;
Convert Binary_Codes to Text_File;
Unzip Text_File;
Output Secret_Message;
End

Performance Measurements

The challenge of using Steganography in cover images is to hide as much data as possible with the least noticeable difference in the stego-image. A tractable objective measures for this property are the Mean Squared Error (MSE) and the Peak-Signal-to-Noise Ratio (PSNR) between the cover image and the stego image. Mean Square Error (MSE): It is the measure used to quantify the difference between the initial and the distorted or noisy image.

Where X and Y are the image coordinates, M and N are the dimensions of the image, S_{xy} is the generated stego-image and C_{xy} is the cover image. From (MSE) we can find Peak Signal to Noise Ratio (PSNR) which measures the quality of the image by comparing the original image with the stego-image. (PSNR) is used to evaluate the quality of the stego-image after embedding the secret message in the cover. It is computed using the following formula:

$$PSNR = 10log_{10}\left(\frac{C^2max}{MSE}\right)$$

Where, *C max* holds the maximum value in the image that is 255. Finally, other associated measures are the Steganographic capacity, which is the maximum information that can safely embedded in a work without having statistically detectable objects. An important note is that, for all the cover images, PSNR are more than 37 dB, this means that the proposed Steganography algorithm provides very good imperceptibility performance and the stego images can't be detected.

IV. PROPOSED ANALYSIS

LSB and RDH Algorithm Implementation Using Steganography

Data hiding is referred to as a process to hide data (representing some information) into cover media. That is, the data hiding process links two sets of data, a set of the embedded data and another set of the cover media data. The relationship between these two sets of data characterizes different applications. For instance, in covert communications, the hidden data may often be irrelevant to the cover media. In authentication, however, the embedded data are closely related to the cover media. In these two types of applications, invisibility of hidden data is an important requirement. In most cases of data hiding, the cover media will experience some distortion due to data hiding and cannot be inverted back to the original media.

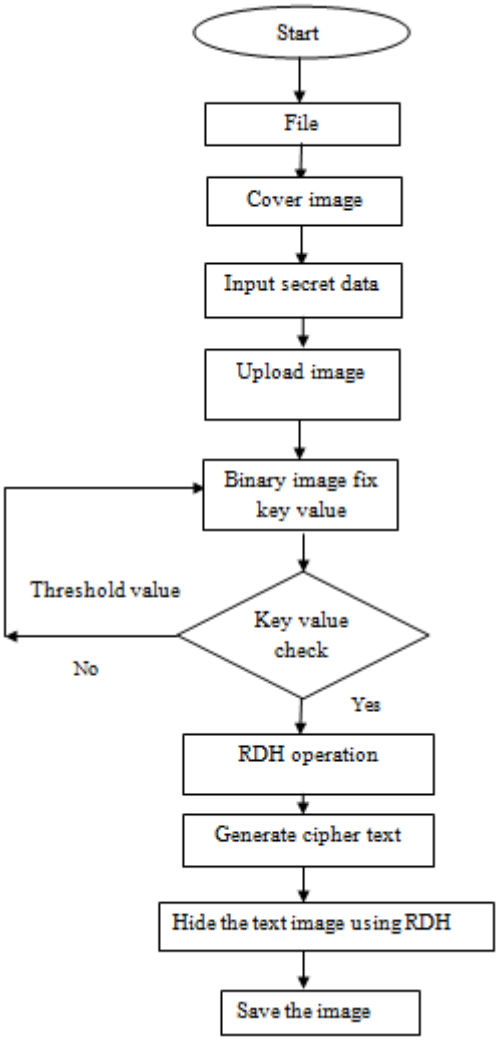

Fig : Flow Chart of Encryption and Embedding Crypto-Steganography

Data hiding Encryption

For an M x N image, each pixel grayscale value $x \in [0, 255]$

1. Generate its *H(x)*
2. In the H(x), find the maximum point *h(a). a* $\in [0, 255]$ and Minimum point h(b), $b \in [0, 255]$
3. If the minimum point *h(b) > 0,* recode the coordinate *(i,j)* of those pixels and the pixels grayscale value b as overhead book keeping information. Then set h(b) =0.
4. Without loss of generality, assume $a < b$. move the whole part of the *H(x)* with x $\in (a, b)$ to the right by 1 unit . this means that all the pixel grayscale values (satisfying x $\in (a, b)$ are added by 1.
5. Scan the image, once meet the pixel (whose grayscale value is a), check the to be embedded bit. If the to be embedded bit is "1", the pixel gray

scale value is changed to a+1. If the bit is "0", the pixel value remains a.

Algorithm to Embedded the text message:

Step 1: Read the cover image and the text message which is to be hid den in the cover image.
Step 2: Convert the text message in binary format.
Step 3: Calculate the LSB of each pixel of the cover image.
Step 4: Replace the cover image of the LSB with each bit of secret message one by one.
Step 5: Write stego image
Step 6: Calculate the Mean square Error (MSE) and the Peak signal to noise ratio (PSNR) of the stego image.

Data Encryption Algorithm

Step 1: Extract the pixels of the cover image.
Step 2: Extract the characters of the text.
Step 3: Extract the characters from the Stego key.
Step 4: Choose first pixel and pick characters of the Stego key and place it in first component of pixel.
Step 5: Place some terminating symbol to indicate end of the key. 0 has been used as a terminating symbol in this algorithm.
Step 6: Insert characters of text in each rgb component of next pixels by replacing it.
Step 7: Repeat step 6 till all the characters has been embedded.

Algorithm to Stego retrieves the text message

Step 1: Read the cover image and text message that has to be hidden in the cover image.
Step 2: Convert the text message into binary. Apply 2D Haar transform on the cover image.
Step 3: Obtain horizontal and vertical filtering coefficients of the cover image and the cover image is then added with data bits for DWT coefficients.
Step 4: Obtain stego image.
Step 5: Calculate the Mean square Error (MSE) and Peak signal to noise ratio (PSNR) of the stego image.

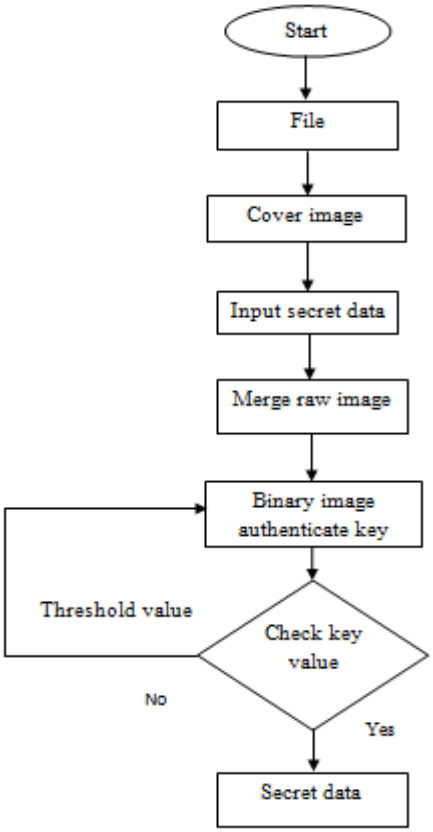

Fig : Chart of Decryption and Extraction Crypto-Steganography

In this framework, a content owner encrypts the original image using a standard cipher with an encryption key. After producing the encrypted image, the content owner hands over it to a data hider (e.g., a database manager) and the data hider can embed some auxiliary data into the encrypted image by losslessly vacating some according to a data hiding key.

Least Significant Bit Hiding Technique (LSB)

LSB is the most popular Steganography technique. It hides the secret message in the RGB image based on it its binary coding. An example about pixel values and shows the secret message. LSB algorithm is used to hide the secret messages by using algorithm 1. LSB makes the changes in the image resolution quite clear as well as it is easy to attack.

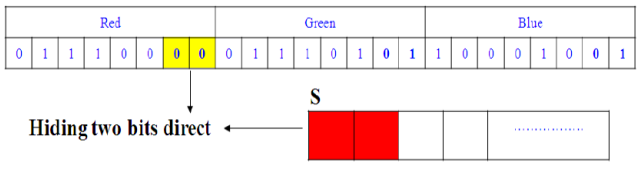

Fig: Least Significant Bit Hiding Technique

LSB hiding technique hide the secret message directly in the least two significant bits in the image pixels, hence that affect the image resolution, which reduce the image quality and make the image easy to attack. As well as this method is already has been attacked and broken. Therefore a new technique that able to make the secret message more secure and enhance the quality of the image.

New method in image Steganography

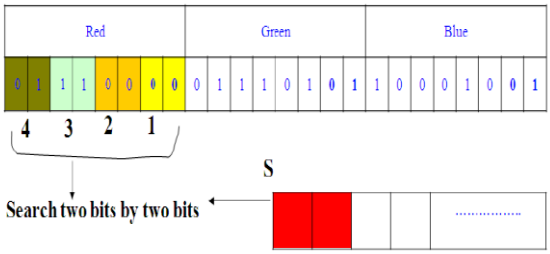

Fig. 4.4 Least Significant Bit Hiding Technique

The message retrieving is done as per the algorithm given below

Step 1) Extract Red, Green and Blue Components of pixel intensity values of Stego image.
Step 2) Take successive Red, Green and Blue component values of
pixels and convert them into array of values for message and Stego image.
Step 3) Convert every decimal value into 8 bit binary equivalent for Stego image.
Step .4) Retrieval of the message bit is done by using the XOR operation on the LSB and Next to LSB.

1. If it is 1 then message bit is 1.
2. If it is 0 then message bit is 0.

Step 5) If during embedding the LSB 3 method is used then retrieval is done by performing XOR operation on LSB, next to LSB and Next to Next to LSB‟s.

1. If the result of XOR operation is 0, it means the decoded message bit value is 0 and
2. If the result is 1, it means that the decoded message bit value is 1.

Step 6) Convert every 8 bits to form a byte whose decimal value is the pixel intensity if the message embedded is a gray scale image otherwise this decimal value forms the intensity of Red Component of the first pixel of the Secret image, if the message embedded is a 64 bit color image. If the message that is embedded is a text message then after every 7 bits are Retrieved convert them into decimal which forms the ASCII code of the 1st character. In this manner these steps are continued till the full message is retrieved.

V. EVALUATION RESULT

For Experimental purpose, 2 sets (24 bit color) of fifteen cover images each, one set of size 512 X 512 and the other of size 1024 X 1024 for all the three methods (LSB 2, LSB 3 and Parity) are used. Ten secret message images and 4 Microsoft word documents are used for embedding. The 10 message images are 128X128, 24-bit color images and out of the four Microsoft word documents 1st is 1262 words (6764 characters), 2nd message is the famous 396 words (1988 characters), the 3rd is a 4171 words, and 23496 characters, and the 4th 78 words and 486 characters. All cover images, message images and text messages are given in appendix. The Experimental results obtained for some of the Cover images, Stego Images. It represents 24 bit cover images of size 512 X 512. stego images using LSB 2 XOR method, stego images using LSB 3 XOR method stego images using LSB parity method respectively with message image 1 embedded in it. are Cover images of size 1024 X 1024. Stego images using LSB 2 bit XOR algorithm are Stego images using LSB 3 bit XOR algorithm are Stego images using LSB parity method with message 1 embedded in it.

For performance evaluation of different methods the parameters used are Mean Squared Error (MSE), Peak Signal to Noise Ratio (PSNR), Average Fractional Change in Pixel Value (AFCPV), Percentage of Bytes Changed, Percentage change in LSB‟s (from Zero to One and One to Zero

Mean Square Error (MSE)

The MSE represents the cumulative squared error between the decompressed/reconstructed and the original image. MSE between two images can be computed as

$$MSE = \sum_{MN}\left(\frac{(I1(M,N) - I2(M,N2)}{M*n}\right)$$

1. M and N : number of rows and columns in the input images, respectively.
2. I1 (m,n) :image pixel value at position (m,n) in the original image
3. I2(m,n) is the image pixel value at position (m,n) in the stego image.
4. Lesser the MSE, better the quality of reconstructed image.

Peak Signal to Noise Ratio (PSNR)

PSNR represents a measure of the peak error. The higher the PSNR, the better the quality of the decompressed or reconstructed image. PSNR is computed as,

$$PSNR = 10\ log_{10}\left(\frac{r\ 2}{MSE}\right)$$

R is the maximum fluctuation in the input image data type. Fo example, if the input image has a double-precision floating-point data type, then R is 1. If it has an 8-bit unsigned integer data type, R is 255.

Percentage of Bytes Changed (PBC)

Another useful parameter which is used to measure the quality of the reconstructed image is Percent of Bytes changed. Lesser the value, better the reconstructed image. Percentage of Bytes changes is computed as

$$P = \left(\frac{No\ of\ Bytes\ Changed}{Total\ No\ of\ Bytes}\right)^* 100$$

Retrieval Accuracy (RA)

This parameter is used to measure the retrieval accuracy of the secret message from stego image.

$$P = \left(\frac{No\ of\ correct\ Bytes\ recovered}{Total\ No\ of\ Bytes}\right)^* 100$$

The values of MSE, PSNR, AFCPV, Percentage of bytes changed, Changes (0 to 1) in percent and Changes (1 to 0) in percent for embedding a message image Cover images 16 -30 of size 1024 X 1024 using LSB 2-bit Method. The last row shows their average values for Cover images 16 – 30.

Cover images	Message capacity	PSNR	
		LSB – 3	ZOH
Boat	8,160	39.1132	49.9386
Bird	8,160	39.0955	49.9167
Flinstone	8,160	39.1188	49.9513

Table : Comparison between (LSB-3) and (ZOH) Methods

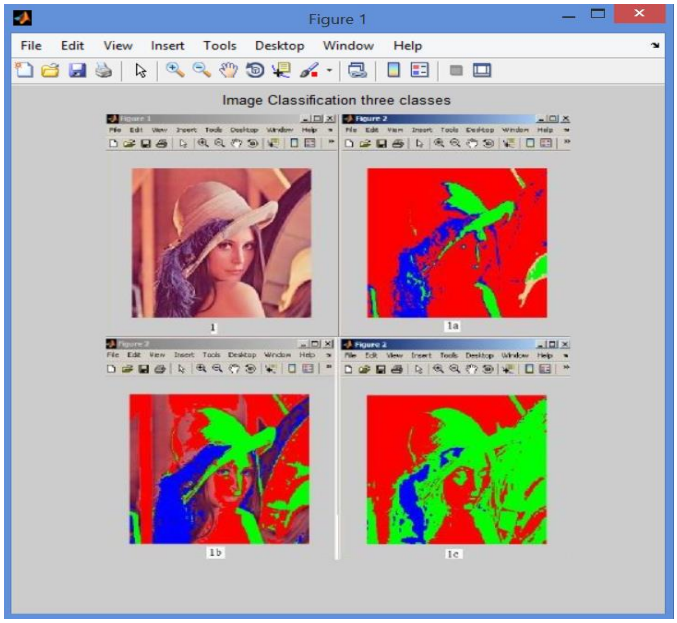

Fig : Image Classification three Classes

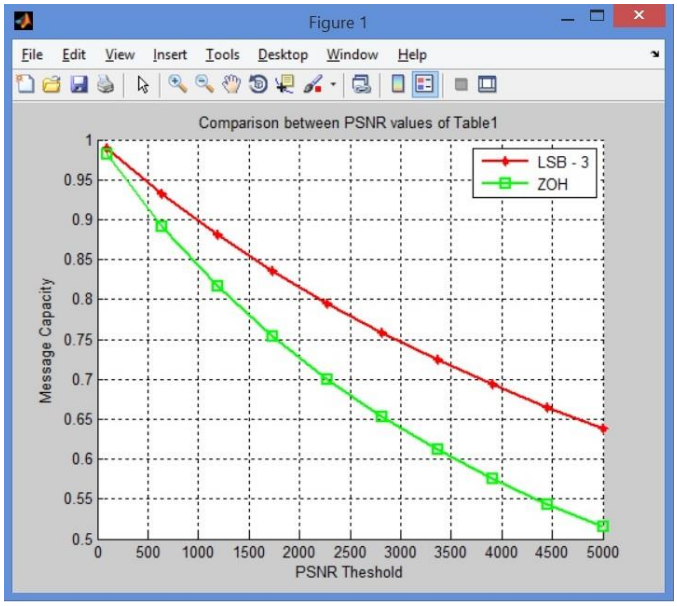

Fig : Comparison between PSNR values of Table1

As shown in Table 1 and Fig 2, after hiding the same message length 8,160 bytes in the cover images (Boat, Bird, Flinstone) with size (256 x 256), using the (LSB-3) and (ZOH) methods, it has been found that, the (ZOH) method has higher PSNR values than the (LSB-3).

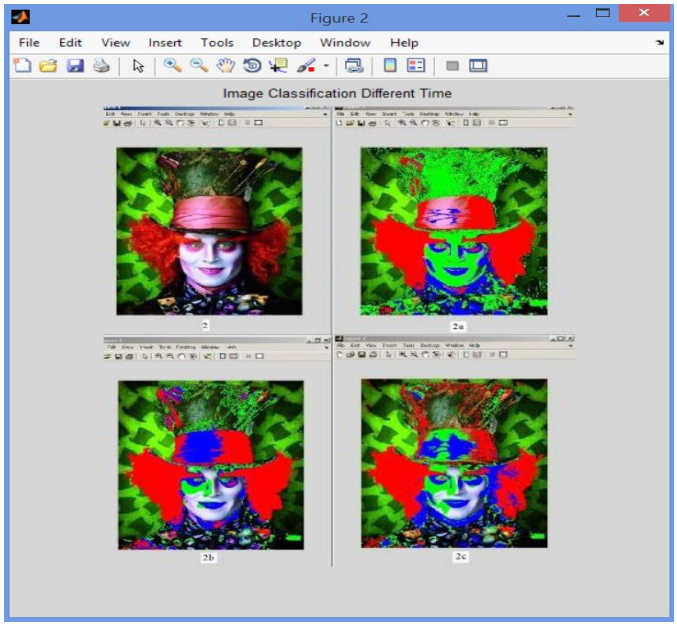

Fig : Image Classification for Different Time

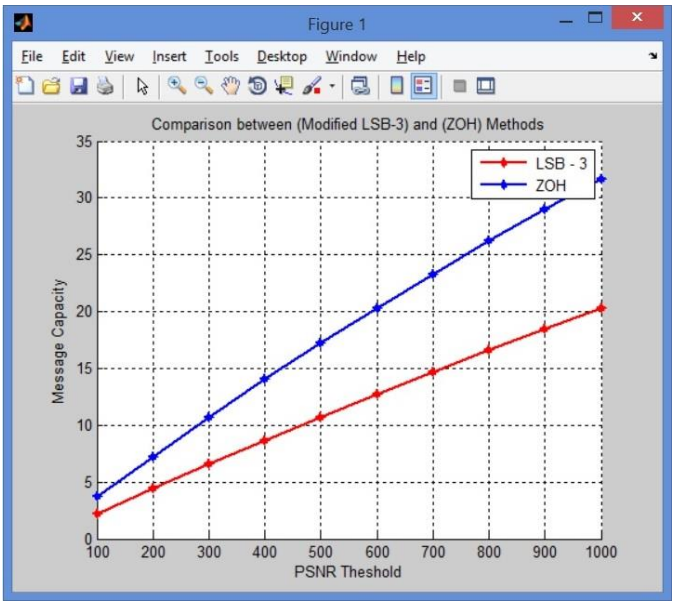

Fig : Comparsion between Modified LSB-3 and ZOH Method

Also in Table 2 and Fig 3, after hiding the same message length 8,160 bytes in the cover images (boat, bird, flinstone) with size (256 x 256), using the (Modified LSB-3) and (ZOH) methods, it has been found that, the (ZOH) method has higher PSNR values than the (Modified LSB-3).

CONCLUSION

Steganography are two important branch of information security. Steganography is the art and science of hiding communication .Steganography involves hiding information so it appears that no information is hidden at all. Image steganography has many applications, especially in today's modern and high-tech world. Steganography have many advantages but it have some limitations also. Privacy and secrecy is a concern for most people on the internet. Image steganography allows for two parties to communicate secretly and covertly. It will focus to develop a high security model for secret data, which uses Steganography. Reversible Data Hiding (RDH) technique is a secure way of transmitting data inside a cover media is used for encryption. as shown in comparison tables, after doing the same experiments using the ZOH, four different methods, the PSNR values of the proposed method (ZOH) were higher than other methods, this means the method's PSNR (stego image quality) is improved. As a future work, we will try to improve Maximum Hiding Capacity (MHC) by improving the secret message using image steganography method (Image Steganography Method by Using Braille Method of Blind People). by representing the secret message characters by using Braille method of reading and writing for blind people that can save more than one-fourth of the required space for embedding.

FUTURE WORK

There is two different methods for lossless data embedding for JPEG images. The first method is based on compression of LSBs of a selected quantized DCT coefficient from all blocks. The second method uses a special trick to preprocess the image to allow trivially lossless data embedding. It is based on manipulation of the quantization table. Finally, it describe three important applications of the lossless data embedding – lossless authentication of images, detection of LSB steganography in images, and lossless robust watermarking. It developed method is considered an effective method which achieved high level of capacity, higher PSNR for security and lower MSE for robustness against attacks. As A new image steganography method will be developed so frequency domain like MSE Algorithm will be used instead of special domain and develop the cryptography method by using asymmetric method like RSA method to enhance the security.

REFERENCES

1. Bret D., "A detailed look at Steganographic Techniques and their use in an Open-Systems Environment", Sans Institute, 1(2002).
2. Akbas E. A., "A New Text Steganography Method By Using Non Printing Unicode Characters", Eng. & Tech. Journal, VOL.28, NO.1, 2010.
3. http://www.tutorialspoint.com/dip/Zooming_Methods.htm.
4. Por L. Y., Delina B., "Information Hiding: A New Approach In Text Steganography" 7th WSEAS int.

Conf. on Applied Computer & Applied Computational Science (ACACOS '08), Hangzhou, China, April 6-8, 2008

5. Abdelmged A. A., Al-Hussien S. S., "Image Steganography Technique By Using Braille Method of Blind People (LSBraille) ", International Journal of Image Processing (IJIP), Vol 7, Issue 1, 2013.
6. Chutani S., Goyal H. "LSB Embedding In Spatial Domain - A Review of Improved Techniques". International Journal of Computers & Technology, ISSN: 2277-3061, 3(1), Aug. 2012.
7. Atallah M. A. "A New Method in Image Steganography with Improved Image Quality". Applied Mathematical Sciences, 6(79), pp. 3907 – 3915, 2012.
8. Aiad, I. A., "Hiding Data Using LSB - 3 ", J.Basrah Researches (Sciences), Vol. 33, No.4. (81-88), December, 2007.
9. Ahmed A. R., Ahmed S. and Al-Hussien S. S., " A High Capacity SLDIP (Substitute Last Digit In Pixel ", Fifth International Conference on Intelligent Computing and Information Systems (ICICIS 2011), 30 June - 3 July, 2011, Cairo, Egypt.
10. Johnson, Neil F., "Steganography", 2000, URL: http://www.jjtc.com/ stegdoc/index2.html
11. Jagvinder K. and Sanjeev K., "Study and Analysis of Various Image Steganography Techniques", IJCST Vol. 2, Issue 3, September 2011.
12. Al-Hussien S. S., " Enhancing the (MSLDIP) Image Steganographic method (ESLDIP Method) ", International Conference on Graphic and Image Processing (ICGIP 2011), Proc. of SPIE Vol. 8285, 82853I, © 2011 SPIE.
13. Mohammed A. F., "Image Steganography by Mapping Pixels to Letters ", Journal of Computer Science, 5 (1): 33-38, ISSN 1549-3636, 2009.