Image Steganography Scheme Based on Reversible Data Embedding Strategy

U. T. Tilakaratne*
Postgraduate Institute of Science
University of Peradeniya
Peradeniya, Sri Lanka
udeshikat@gmail.com

U. A. J. Pinidiyaarachchi
Department of Statistics and Computer Science
Faculty of Science, University of Peradeniya
Peradeniya, Sri Lanka
ajp@pdn.ac.lk

***Abstract*—Steganography is communicating in a way which hides the existence of the information. The cover media can be any general digital format, such as, text, image, audio, video, etc. By using reversible data embedding strategy, the original digital format can be restored. In this paper, we present a reversible data embedding scheme with a good stego image quality and high capacity, which is applicable for any image type and any image format. Experimental results show that the average PSNR value is greater than 60dB and the hiding capacity in RGB images are three times higher than the grayscale images.**

***Index Terms*—Image Steganography, Reversible data embedding, Inverse embedding, PSNR**

I. Introduction

Steganography deals with embedding information in a cover media without making any visible changes [1]. The cover media can be any general digital format, such as, text, image, audio, video, etc. In case of image, the image that is used to carry the secret data is referred to as the cover image and the image that carried the secret data is referred to as stego image.

Two main factors that really affect an embedding scheme are visual quality of stego images and embedding capacity (or payload). A steganographic method with low image distortion is more secure than that with high distortion because it does not raise any suspicions of adversaries. The second important factor is embedding capacity. A steganographic method with high payload is preferred because more secret data can be transferred. Thus, the hiding capacity should be as high as possible under the condition that cannot be distinguished by human vision.

Reversible (or Lossless) data hiding is a technique that not only embeds data into cover images, but also restores the cover images from the stego image after the secret data have been extracted [2]. In applications, such as in law enforcement, medical image systems, it is desired to be able to reverse the stego-media back to the original cover media for legal consideration. In remote sensing and military imaging, high accuracy is required. In some scientific research, experimental data are expensive to be achieved. Under these circumstances, the reversibility of the original media is desired.

In this paper, we present a high capacity and low distortion reversible data embedding method for digital images. Our method is an improved version of the method proposed by L. Jia, S. H. Shin and K. Y. Yoo [3] in 2010, a reversible data hiding scheme using inverse embedding methods in double-embedding strategies.

II. Existing Methods

A simple method of data hiding involves the manipulation of the least significant bit (LSB) plane of the data. Various techniques, such as direct and random replacement of the cover LSBs with message bits or an arithmetic combination between the two are used. Several examples of LSB schemes can be found in [4],[5] and [6]. LSB replacement methods typically achieve high payload, but introduce some amount of distortion into the original image and the distortion is permanent and not reversible. Therefore, these methods can only be used when the content of the cover image has no value to the sender or the receiver.

In 2003, Tian [7] proposed a reversible data hiding scheme using a difference expansion (DE). This method embeds a secret bit stream into a grayscale cover image to obtain the stego image. Specifically, the cover image is scanned in raster scan order (i.e., from left to right and top to bottom) to group two neighboring pixels into a pixel pair (x, y). Firstly, the integer average m and the difference value d of x and y are computed by; $m = \lfloor (x+y)/2 \rfloor$ and $d = x - y$. Secondly, the secret bit b is embedded into d by the difference expansion (DE) operation to obtain the new difference value d' as, $d' = 2 \times d + b$. Finally, the embedded pixel pair (x', y') is calculated by $x' = m + \lfloor (d'+1)/2 \rfloor$, $y' = m - \lfloor d'/2 \rfloor$. In DE scheme, the maximum hiding capacity is no more than 0.5bpp, but the visual quality is not maintained very well [3].

In 2009, Duc Kieu and Chin-Chen Chang [8] proposed a new scheme with reversibility with high image quality and high payload using multiple embedding strategy. This scheme consists of two main stages, the horizontal embedding procedure and the vertical embedding procedure, which are used to embed the secret data, and the LSB replacement method is used to embed the compressed location maps. Duc Kieu scheme is proposed for grayscale images and is capable

978-1-4673-4463-0/13/$31.00 ©2013 IEEE

of achieving a good visual quality of stego images with a high embedding capacity especially when multiple layer embedding is performed. But in this scheme they have embed location maps to the image using LSB replacement, therefore the complete reversibility of the image is not guaranteed.

In 2010, L. Jia, S. H. Shin and K. Y. Yoo [3] proposed a high hiding capacity reversible data hiding scheme with low distortion using inverse embedding methods in double-embedding strategies for 8-bit gray scale images. In this scheme Embedding phase is divided into two sub phases. In the first phase, select a cover pixel pair (x,y), if the value of y is even, it can be used to embed the secret data. That is if secret data is 1, value of y is increased by 1. In the second embedding phase, select pixel pair (x',y'). If the value of y' is odd, it can be used to embed secret data. That is if the secret data is 0, value of y' is decreased by 1. In the embedding phases location maps (L1 and L2) are used to keep track of embeddable pixels. Inverse of the first embedding phase is used in the second embedding phase to increase the hiding capacity and to decrease the distortion at the same time.

The experimental results demonstrate that this scheme has a very good performance in both hiding capacity and stego image quality but only for 8-bit grayscale images. In this paper we have improved the Jia's scheme to work for grayscale images and RGB images, and all commonly used image formats like .bmp, .jpg, .png, .giff and .tiff.

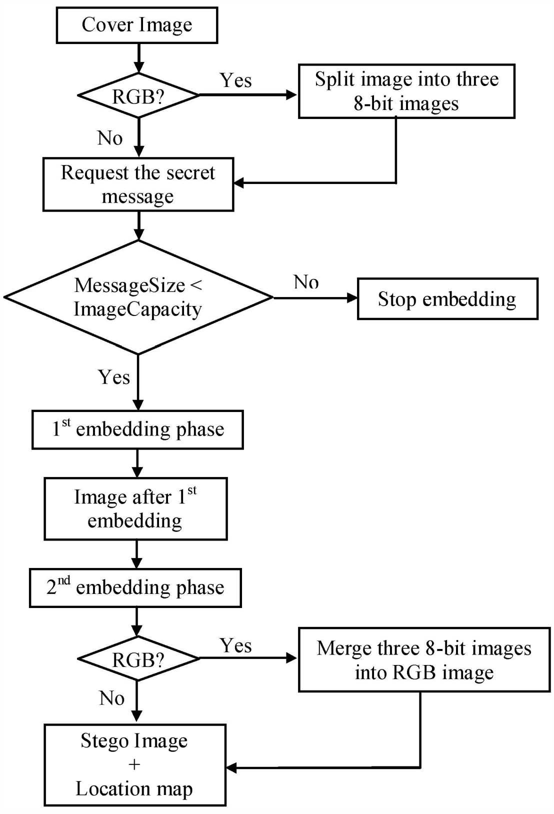
III. METHODOLOGY

Reversible or lossless embedding technique enables the exact recovery of the original image upon extraction of the embedded information. Location maps are used to identify or locate the changeable pixel values.

The proposed scheme uses one pixel pair to embed one bit secret data, and the additional location maps are generated in the embedding phase and used to reconstruct the cover image in the extraction phase.

Embedding phase

- Let ***I*** be the cover Image with size ***wxh*** (where *w* is the width of the image and *h* is the height), and $I_{i,j}$ be the pixel located on row ***i***, column ***j*** in image *I*, where $0 \le i \le w - 1$, $0 \le j \le h - 1$. *I"* is the stego image and its size is the same as the size of *I*.
- The secret data is denoted by **S**, S = $s_0 s_1 s_2 \ldots\ldots s_{l-1}$, where $s_k \in \{0,1\}$, $0 \le k \le l - 1$, and *l* is the length of S.
- Location map is a dynamic array and is denoted by **L**. It used to locate the changeable pixels of the cover image. Size of the location map is always less than h*w/2.
- Payload (or capacity) of the image is calculated during the embedding phase. Size of the message should be less than total capacity.



Cover Image
RGB?
Yes
Split image into three 8-bit images
No
Request the secret message
MessageSize < ImageCapacity
No
Stop embedding
Yes
1st embedding phase
Image after 1st embedding
2nd embedding phase
RGB?
Yes
Merge three 8-bit images into RGB image
No
Stego Image + Location map

Figure 1 - Proposed Algorithm – Embedding Phase

Figure 1 shows the flow chart of the embedding phase. The steps in the embedding phase are listed as follows:

Input: Cover image *I*, secret data S

Output: Stego image *I"*, location map L

Step 1: Check the cover image format:

(1.1) If image is a grayscale image go to *Step 2*.

(1.2) If the cover image is a RGB color image, first split the image into red, green and blue 8-bit images. Then go to *Step 2*.

Step 2: Divide the secret message S into two messages (S1 and S2) and convert to binary. Then scan the image *I* to get the first pixel pair ($I_{i,j}$,$I_{i+1,j}$), *i*=*j*=0.

Step 3: Check whether S1 can be embedded to the image or not.

(3.1) Count number of even pixels countEven

(3.2) If *l*1≤ countEven go to *Step 4*, where *l*1 is the length of S1.
Otherwise, stop the embedding process.

Step 4: Select the even pixel value to embed the secret data:

(4.1) If, $I_{i+1,j}$ is odd, L_k=0 and then set $I'_{i,j} = I_{i,j}$, $I'_{i+1,j} = I_{i+1,j}$ and if $i = w - 2$, then set *j*=*j*+1 and *i* = 0, otherwise update *i* by *i*+2, then get the next pixel pair ($I_{i,j}$,$I_{i+1,j}$) and go to *Step 4.*

(4.2) If $I_{i+1,j}$ is even, this pixel pair can be used to embed the secret data, L_k=1 and then $I'_{i,j}$=$I_{i,j}$, and go to *Step 5*.

Step 5: Embed the secret bit by the following equation:

$$I'_{i+1,j} = \begin{cases} I_{i+1,j} & if\ S_k = 0 \\ I_{i+1,j} + 1 & if S_k = 1 \end{cases}$$

If $i = w - 1$ and $j = h - 1$, go to *Step 6*;
Otherwise if $i = w - 2$, then set $j=j+1$ and $i = 0$,
Otherwise update i by $i+2$, then get the next pixel pair $(I_{i,j}, I_{i+1,j})$ and go to *Step 4.*

Step 6: Scan the image I' to get the first pixel pair $(I'_{i,j}, I'_{i+1,j})$, $i=j=0$.

Step 7: Check whether S2 can be embedded to the image or not.

(7.1) Count number of odd pixels countOdd

(7.2) If $l2 \leq$ countOdd go to *Step 8*, where $l2$ is the length of S2.
Otherwise, stop the embedding process.

Step 8: Select the odd pixel value to embed the secret data:

(8.1) If, $I'_{i+1,j}$ is even, $L_k=0$ and then set $I''_{i,j} = I'_{i,j}$, $I''_{i+1,j} = I'_{i+1,j}$ and if $i = w - 2$, then set $j=j+1$ and $i = 0$, otherwise update i by $i+2$, then get the next pixel pair $(I'_{i,j}, I'_{i+1,j})$ and go to *Step 8.*

(8.2) If $I'_{i+1,j}$ is odd, this pixel pair can be used to embed the secret data, $L_k=1$ and then $I''_{i,j} = I'_{i,j}$, and go to *Step 9.*

Step 9: Embed the secret bit by the following equation:

$$I''_{i+1,j} = \begin{cases} I'_{i+1,j} - 1 & if\ S_k = 0 \\ I'_{i+1,j} & if S_k = 1 \end{cases}$$

If $i = w - 1$ and $j = h - 1$, go to *Step 10*;
Otherwise if $i = w - 2$, then set $j=j+1$ and $i = 0$,
Otherwise update i by $i+2$, then get the next pixel pair $(I_{i,j}, I_{i+1,j})$ and go to *Step 8.*

Step 10: If grayscale image go to *Step 11.* Otherwise merge 8-bit red, green and blue images and go to *Step 11.*

Step 11: Output the stego image I'' and the location map L.

Extraction Phase

Figure 2 shows the flow chart of the extraction phase, and the steps in the extraction phase are listed as follows:

Input: Stego image I'', Location map L.

Output: Cover image I, Secret data S

Step 1: Check the cover image format:

(1.1) If image is a grayscale image go to *Step 2.*

(1.2) If the cover image is a RGB color image, first split the image into red, green and blue 8-bit images. Then go to *Step 2.*

Step 2: Scan the image I'' to get the first pixel pair $(I''_{i,j}, I''_{i+1,j})$, $i=j=0$.

Step 3: If $L_k=0$, then set $I'_{i,j} = I''_{i,j}$, $I'_{i+1,j} = I''_{i+1,j}$ and if $i = w - 2$, then set $j=j+1$ and $i = 0$, otherwise update i by $i+2$, then get the next pixel pair $(I''_{i,j}, I''_{i+1,j})$ and go to *Step 3.*

If $L_k=1$, then this pixel pair contains the secret data, go to *Step 4.*

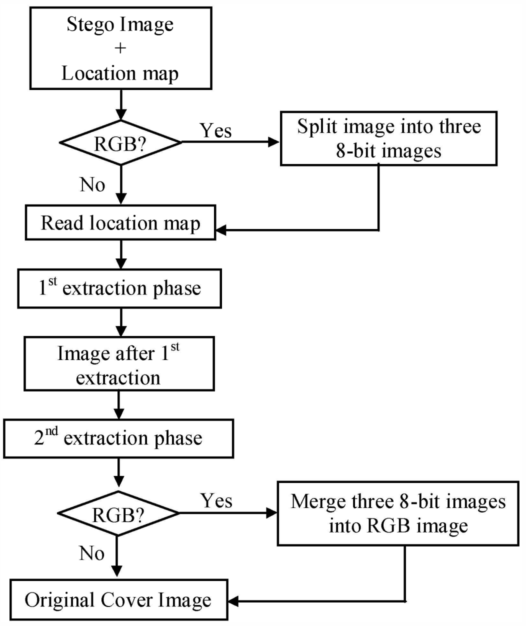


Figure 2 - Proposed Algorithm – Extraction Phase

Step 4: Extract the secret data:

(4.1) If, $I''_{i+1,j}$ is even, extract the secret data $S_k = 0$, and reconstruct the cover image by the following equation:

$$I'_{i+1,j} = I''_{i+1,j} + 1$$

(4.2) If $I''_{i+1,j}$ is odd, extract the secret data $S_k=1$, and reconstruct the cover image by the following equation:

$$I'_{i+1,j} = I''_{i+1,j}$$

If $i = w - 1$ and $j = h - 1$, go to *Step 5*;
Otherwise if $i = w - 2$, then set $j=j+1$ and $i = 0$,
Otherwise update i by $i+2$, then get the next pixel pair $(I_{i,j}, I_{i+1,j})$ and go to *Step 3.*

Step 5: Scan the image I' to get the first pixel pair $(I'_{i,j}, I'_{i+1,j})$, $i=j=0$.

Step 6: If $L_k=0$, then set $I_{i,j} = I'_{i,j}$, $I_{i+1,j} = I'_{i+1,j}$ and if $i = w - 2$, then set $j=j+1$ and $i = 0$, otherwise update i by $i+2$, then get the next pixel pair $(I'_{i,j}, I'_{i+1,j})$ and go to *Step 6.*

If $L_k=1$, then this pixel pair contains the secret data, go to *Step 7.*

Step 7: Extract the secret data:

(7.1) If, $I'_{i+1,j}$ is even, extract the secret data $S_k = 0$, and reconstruct the cover image by the following equation:

$$I_{i+1,j} = I'_{i+1,j}$$

(7.2) If $I'_{i+1,j}$ is odd, extract the secret data $S_k=1$, and reconstruct the cover image by the following equation:

$$I_{i+1,j} = I'_{i+1,j} - 1$$

If $i = w - 1$ and $j = h - 1$, go to *Step 8*;
Otherwise if $i = w - 2$, then set j=j+1 and i = 0,
Otherwise update i by i+2, then get the next pixel pair ($I_{i,j}$,$I_{i+1,j}$) and go to *Step 6.*

Step 8: Output the reconstruct cover image *I*, and the secret data S.

IV. EXPERIMENTAL RESULTS

To evaluate the performance, we implemented the proposed method using ImageJ and tested it using different image types and image formats of "Lena" of sized 512×512 (Figure 3). Embedded message is the introduction section of this research paper, which consists of 1837 characters.

PSNR (Peak Signal to Noise Ratio) is a common way of assessing the quality of digital images. It is used to measure the distortion between the cover image and the stego image. The hiding capacity is measured by counting the number of bits that can be stored in each image. Table 1 shows the PSNR values and hiding capacity for different image types and formats. All PSNR values are greater than 60dB, which means distortion is comparatively less. But there is a visible distortion in 8bit color "Lena.gif" image as shows in Figure 3. The hiding capacities in RGB images are higher than those of grayscale images. Therefore we can store more information in RGB images.

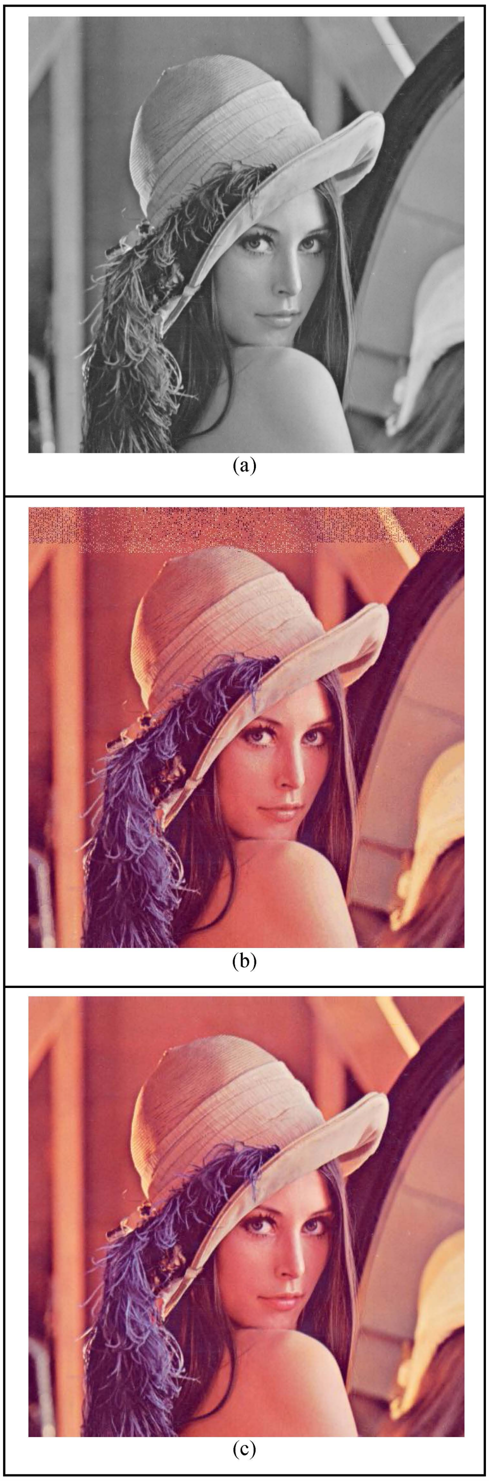
The proposed method increases or decreases the cover pixels by 1. Specifically, first embedding phase increases the cover pixels by 1 and the second embedding phase decreases the cover pixels by 1. Among the cover pixels modified by the first embedding phase, many of them have a chance to possibly get back to their original values in the second embedding phase. This fact can be visually observed through analyzing histograms of original cover images and stego images as shown in Figure 4.

In order to maintain the complete reversibility, location map is not embedded into stego image. It is shared with the receiver as the secret key for data extraction.

To evaluate the performance in relation to increasing message size, several message sizes have been used and the PSNR values were obtained. Figure 5 shows the embedded message size and the corresponding PSNR of RGB image "Lena.bmp". By analyzing it we can see that the average PSNR value is greater than 60dB.

TABLE I. THE PERFORMANCE RESULTS OF THE PROPOSED METHOD

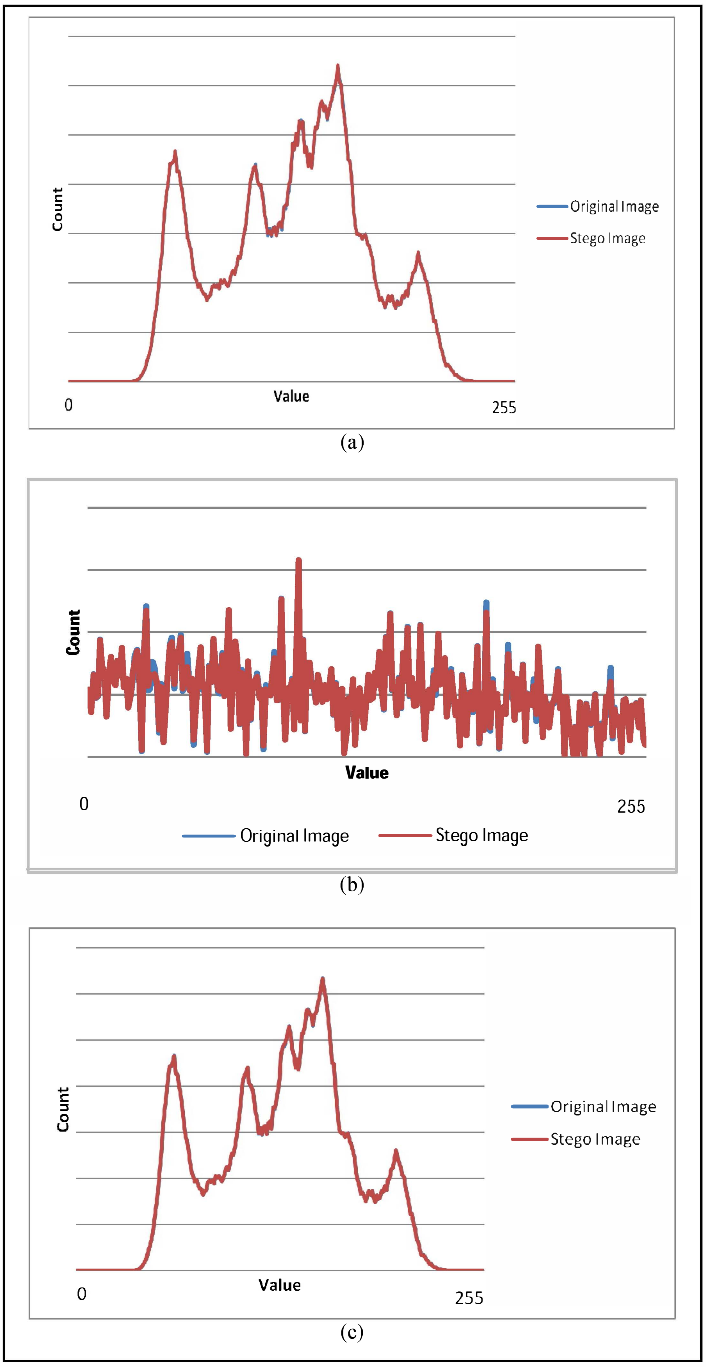
Image Type and Format	PSNR (dB)	Hiding Capacity (No of bits)
8-bit Grayscale / .tiff	64.796	131 kb
16-bit Grayscale/ .tiff	64.796	131 kb
8-bit Color / .gif	27.497	131 kb
RGB Color/ .bmp	70.496	393 kb
RGB Color / .png	70.496	393 kb
RGB Color / .tiff	70.496	393 kb
RGB Color / .jpg	70.665	393 kb


(a)

(b)

(c)

Figure 3 – Stego images. (a)Lena.bmp 8bit grayscale, (b) Lena.gif 8bit Color, (c) Lena.tiff RGB.



(a)

(b)

(c)

Figure 4 – Histograms. (a) Lena.bmp 8bit grayscale, (b) Lena.gif 8bit Color, (c) Lena.tiff RGB

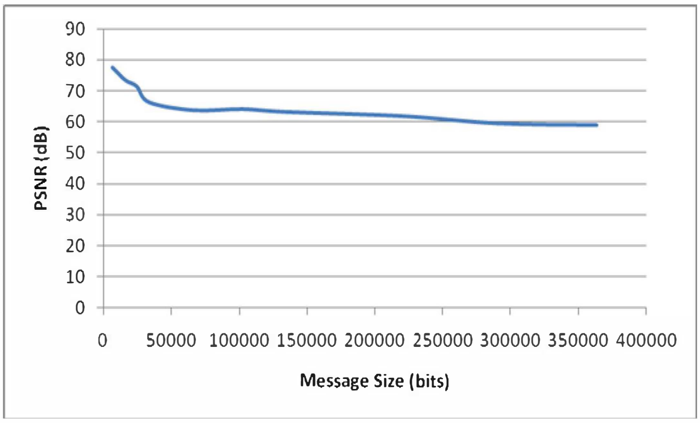


Figure 5 – Message Size Vs PSNR

CONCLUSION

In this paper, we have presented an image steganography scheme based on reversible data hiding strategy that works for all commonly used image types. The experimental results show that the proposed method increases the hiding capacity while decreasing the distortion of the stego image by using the inverse embedding strategy in second embedding phase. However, 8-bit color images are not suitable for proposed steganographic scheme. Future work includes improvement to this scheme to support for 8-bit color images and for audio and video data.

REFERENCES

[1] N. F. Johnson, S. Jajodia, “Exploring Steganography: Seeing the unseen”. IEEE Computer, February 1998, pp.2634

[2] J. Fridrich, M. Goljan and D. Rui., “Lossless Data Embedding - New Paradigm in Digital Watermarking”, In Special Issue on Emerging Applications of Multimedia Data Hiding, Vol. 2, pp. 185-196, February 2002.

[3] L. Jia, S. H. Shin, K. Y. Yoo, “A reversible data hiding scheme using inverse embedding methods in double-embedding strategies”. Informatics and Systems (INFOS), 2010 The 7th International Conference,March 2010, pp.1-7.

[4] R. Rana, D. Singh, “Steganography-Concealing Messages in Images Using LSB Replacement Technique with Pre-Determined Random Pixel and Segmentation of Image”, International Journal of Computer Science and Communication, Dec 2010,1,2, pp.113-116.

[5] A. A. Nikoukar, “An Image Steganography Method with High Hiding Capacity”, International Journal of Signal and Image Processing 2010,1,4, pp. 238-241.

[6] M. Juneja, P. S. Sandhu, E. Walia, “Application of LSB Based Steganographic Technique for 8-bit Color Images”, World Acadamy of Science, Engineering and Technology, 2009,50.

[7] J. Tian, “Reversible data embedding using a difference expansion”, IEEE Transactions on Circuits and Systems for Video Technology, 13(8):pp890-896, August 2003.

[8] D. Kieu, C. C. Chang, “A high stego-image quality steganographic scheme with reversibility and high payload using multiple embedding strategy”, The Journal of Systems and Software 82 (2009), pp.1743-1752.