http://dx.doi.org/10.14257/ijsip.2014.7.6.22

Secure Data Transmission through RDH

S. Bhavani and B. Raviteja

M. Tech in Digital Electronics & Communication Systems
ammav10@gmail.com, braviteja.22@gmail.com

Abstract

Nowadays, with the rapid growth in information technology more and more images and data are available on the internet. So there is a need to provide some kind of authentication to such important data. When the sender transmits the image to the receiver, there may be intruders present in between who may capture the image. After capturing the image the mage the intruder may view the meaningful content in the image. This may not be the problem in some cases. But if we consider security applications like medical and military images then such distortion is unacceptable. To avoid misuse or loss of information several reversible data hiding techniques (RDH) are implemented. This paper gives details on watermarking, LSB, Histogram and RDH using optimum Thresholding with related results.

Keywords: *Reversible data hiding, LSB, Watermarking, Histogram, Optimum value transfer*

1. Introduction

Reversible data hiding scheme is the technique that allows embedding data inside an image and later the hidden data can be retrieved as required and the exact copy of the original host image is recovered. Some of the traditional reversible data hiding schemes are based on modulo-arithmetic additive and spread-spectrum techniques [1, 2]. Although some of these schemes are robust, the modulo-arithmetic-based reversible data-hiding algorithms have the disadvantage of salt-and-pepper visual artifacts and hinder watermark retrieval [3]. In order to enhance the robustness of the reversible watermarking and reduce the salt-and- pepper visual artifact of the above mentioned schemes, histogram shifting techniques were proposed. In this scheme, the embedding target is replaced by the histogram of a block of the image. A good example of the scheme is the circular interpretation scheme proposed by Vleeschouwer *et al.* [4, 5]. Although this type of data hiding schemes provides a higher quality of the embedded image, the embedding capacity is lower [6]. A different category of data hiding schemes involves methods that losslessly compress a set of selected features from an image and embed the payload in the space saved due to the compression. This type results in higher embedding capacity than the previously mentioned types [7]. Another scheme of this type is the generalized Least Significant Bit (g-LSB) embedding algorithm proposed by Celik *et al.* [8], which is based on grouping the pixels and embedding data bits into the state of each group. Each algorithm explained in following sections.

2. Watermarking Technique

Watermarking technique can be classified into two different types. In the first type the watermark is visible i.e. different logos or text can be inserted which is visible. This technique can be seen in Microsoft MS Word where we apply watermarks on the pages which are visible. The second technique used for applying watermarks to the images, videos, is invisible. This invisible technique is called as digital watermarking. The digital watermarking technique can be used for a wide range of applications, like for providing copyright protection to the films, videos, etc. The digital watermark is a more secured technique because the watermark is not visible, so if the intruder views the content of the image he will not be aware of the watermark which is already present in the image. So if the intruder performs any modification in the image it will be known to the receiver after receiving the image. The receiver after receiving the image will see that the watermark has been changed and will be aware that the image has been modified.

2.1. Algorithm to embed one watermark into other watermark

Inputs: Primary watermark, secondary watermark image.

Steps:

1) Read the primary visual watermark image.

2) Decompose the primary watermark image into cap1, chp1, cvp1, cdp1 bands using daubachesis filter.

3) Read the secondary visual watermark image.

4) Add the secondary watermark into the horizontal DWT coefficient (chp1) of Primary watermark.

5) Apply the IDWT to get the nested watermark image.

6) Calculate the PSNR & MSE of nested watermark with original primary watermark image.

Output: Nested watermark image

2.2. Algorithm to embed nested watermark into cover image:

Inputs: Cover image, nested watermark image

Steps:

1) Read the cover image & nested watermark image.

2) Apply DWT to cover image to obtain approximation, horizontal, vertical, diagonal DWT coefficients, *i.e.*, ca1, ch1, cv1, cd1.

3) Modify the approximation DWT coefficient by adding the nested watermark image as in equation

a. Ca1(i,j)=ca1(i,j)+(α*nested watermark)

b. Where Ca1 & ca1 are the modified & original approximation coefficients and α is a scaling value as set to 0.04.

4) Modify the diagonal DWT coefficient by adding the nested watermark image as in the equation

a. Cd1(i,j)=cd1(i,j)+(α*nested watermark)

5) Where Cd1 & cd1 are the modified & original diagonal coefficients of cover image and α is set to value 0.01.

6) Apply the inverse DWT to obtain the watermarked cover image.

7) Calculate the PSNR & MSE of watermarked cover image with original cover image.

Output: Watermarked cover image.

2.3. Algorithm for Watermark Extraction

Inputs: Original cover image, Watermarked cover image or Attacked Watermarked cover image.

1) Apply two-dimensional DWT, to obtain the first level decomposition of the watermarked cover image, *i.e.*, c1a1, c1h1, c1v1, c1d1

2) Extract the watermark from approximation & diagonal DWT coefficient (c1a1 & c1d1) as per the equation

LLW = (c1a1-ca1)/ α; where α=0.04

HHw = (c1d1-cd1)/ α; where α=0.01

3) Calculate the visual quality of extracted watermark by the Similarity Ratio (SR) between compared images. SR= S/(S+D), where S denotes the number of matching pixel values in compared images, and D denotes the number of different pixel values in compared images.

4) Apply the set of possible attacks to watermarked image to get the attacked image. Calculate for each attack the PSNR of Original and attacked image.

5) Apply two-dimensional DWT, to obtain the first level decomposition of the Attacked image, *i.e.*, c2a2, c2h2, c2v2, c2d2.

6) Extract the watermark from approximation & diagonal DWT (i.e. c2a2 & c2d2) coefficient of attacked image as per the equation

LAW = (c2a2-ca1)/ α; where α=3

HAw = (c2d2-cd1)/ α; where α=1

7) Calculate the visual quality of extracted watermark from attacked image by the Similarity Ratio (SR) between compared images. SR= S/(S+D).

Outputs: Extracted Watermarks from approximation & diagonal coefficients of watermarked cover image & Attacked cover image.

The above algorithm results are shown in following figures. Figure 1 represents water mark images, Figure 2 is the original image, Figure 3 shows the algorithm results at sender and Figure 4 represents reversing algorithm at receiver side. Comparison of original with watermarking image in terms of psnr and capacity are tabulated in Table 1 and 2.

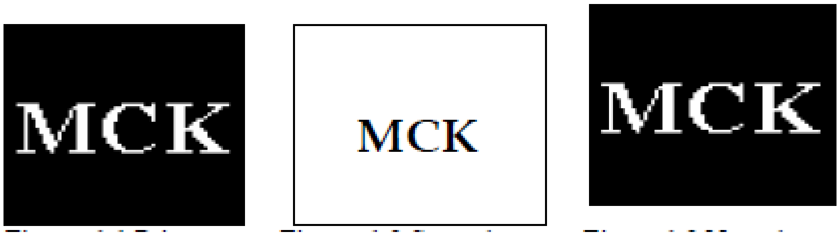


Figure 1. Watermark Images

Figure 2. Original Image

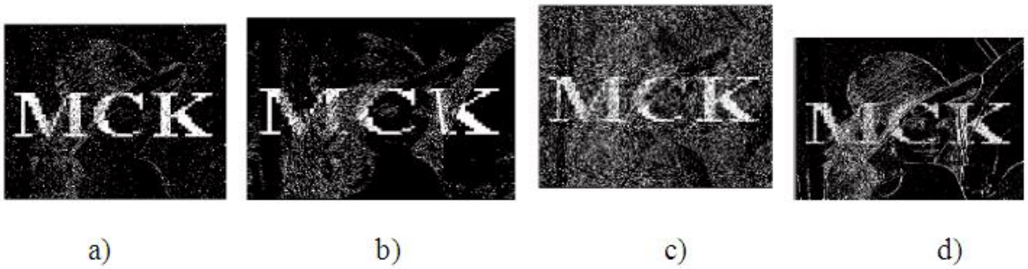


Figure 3. a) Intensity Adjustment; b) Gamma Correction (1.5); c) Histogram Equalization; d) Low Pass Filter (3, 3)

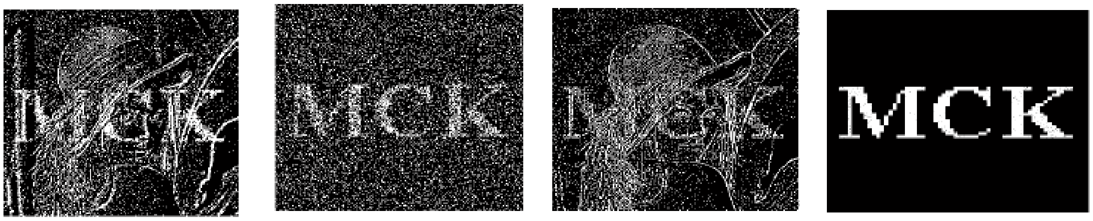


Figure 4: a) Resizing; b) Gaussian Noise (512,256,512) (mean=0, variance 0.001); c) High Pass Filter [0.6]; d) JPEG Compression

Table 1. PSNR1 & PSNR2

Cover image	Primary water mark image	Secondary water mark image	PSNR1	MSE1	PSNR2	MSE2
Lena 512*512	dmg2.tif 64*64	dmg1.tif 64*64	45.15	6.13e-0.06	55.14	3.06e-0.05

Table 2. Capacity Increase Results

Cover image	Capacity of bits without watermarking	Capacity of bits with watermarking
Lena	4096	8091

3. Least-Significant Bit (LSB) Technique

The least significant bit (in other words, the 8th bit) of some or all of the bytes inside an image is changed to a bit of the secret message. Digital images are mainly of two types (i) 24 bit images and (ii) 8 bit images. In 24 bit images we can embed three bits of information in each pixel, one in each LSB position of the three eight bit values. Increasing or decreasing the value by changing the LSB does not change the appearance of the image; much so the resultant stego image looks almost same as the cover image. In 8 bit images, one bit of information can be hidden.

The cover image and stego image are shown in Figure 5 and 6 are obtained by applying LSB algorithm on both the cover and hidden images. The hidden image is extracted from the stego-image by applying the reverse process [9, 10]. If the LSB of the pixel value of cover image C(i,j) is equal to the message bit m of secret massage to be embedded, C(i,j) remain unchanged; if not, set the LSB of C(i, j) to m. The message embedding procedure is given below

S(i,j) = C(i,j) - 1, if LSB(C(i,j)) = 1 and m = 0

S(i.j) = C(i,j), if LSB(C(i,j)) = m

S(i,j) = C(i,j) + 1, if LSB(C(i,j)) = 0 and m = 1

where LSB(C(i, j)) stands for the LSB of cover image C(i,j) and m is the next message bit to be embedded.

S(i,j) is the stego image.

As we already know each pixel is made up of three bytes consisting of either a 1 or a 0.

For example, suppose one can hide a message in three pixels of an image (24-bit colors). Suppose the original 3 pixels are [11]:

(11101010 11101000 11001011)

(01100110 11001010 11101000)

(11001001 00100101 11101001)

A steganographic program could hide the letter ”J” which has a position 74 into ASCII character set and have a binary representation ”01001010”, by altering the channel bits of pixels.

(11101010 11101001 11001010)

(01100110 11001011 11101000)

In this case, only four bits needed to be changed to insert the character successfully. The resulting changes that are made to the least significant bits are too small to be recognised by the human eye, so the message is effectively hidden. The advantage of LSB embedding is its simplicity and many techniques use these methods [12]. LSB embedding also allows high perceptual transparency.

The following Figure 5, 6 &7 shows the mechanism of LSB technique.

3.1. Data Embedding

The embedding process is as follows.

Inputs: Cover image, stego-key and the text file

Output: stego image

Procedure:

Step 1: Extract the pixels of the cover image.

Step 2: Extract the characters of the text file.

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 file in each first component of next pixels by replacing it.

Step 7: Repeat step 6 till all the characters has been embedded.

Step 8: Again place some terminating symbol to indicate end of data.

Step 9: Obtained stego image [13]

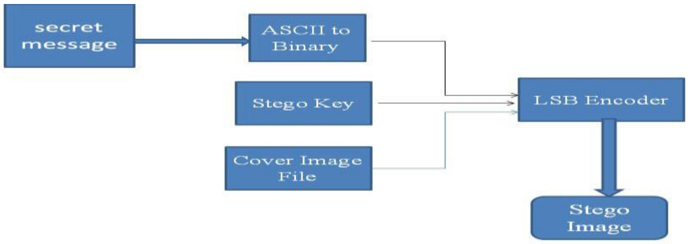


Figure 5. LSB insertion Mechanism

3.2. Data Extraction

The extraction process is as follows.

Inputs: Stego-image file, stego-key

Output: Secret text message. Procedure:

Step 1: Extract the pixels of the stego image.

Step 2: Now, start from first pixel and extract stego key characters from first component of the pixels. Follow Step3 up to terminating symbol, otherwise follow step 4.

Step 4: If this extracted key matches with the key entered by the receiver, then follow Step 5, otherwise terminate the program.

Step 5: If the key is correct, then go to next pixels and extract secret message characters from first component of next pixels. Follow Step 5 till up to terminating symbol, otherwise follow step 6.

Step 6: Extract secret message [14].

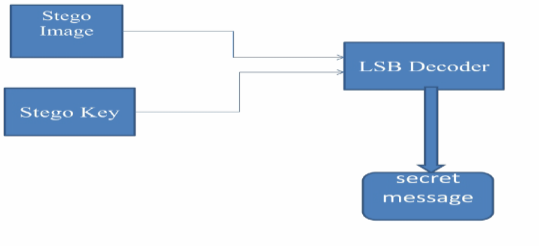


Figure 6. LSB extraction Mechanism

3.3. Image Encoding Algorithm

Inputs: Image file, stego key and image file

Output: Stego image.

1. The cover and secret images are read and converted into the unit8 type.

2. The numbers in secret image matrix are conveyed to 8-bit binary. Then the matrix is reshaped to a new matrix a.

3. The matrix of the cover image is also reshaped to matrix b

4. Perform the LSB technique described above

5. The stego-image, which is very similar to the original cover image, is achieved by reshaping matrix b.

6. While extracting the data, the LSB of the stego image is collected and they are reconstructed into the decimal numbers. The decimal numbers are reshaped to the secret image [15].

Results are tabulated in Table 3 and following figures.

Table 3. PSNR of Least Significant Bits Encoding

SL.No	Cover image	Secret Message	Stego-Image	SNR(dB)	MSE	PSNR(dB)
1	Gray	Text Message	Gray image	58.5044	0.0466	60.4721
2	RGB	Text message	Gray	59.0442	0.0211	67.6897
3	RGB	Image	Hydrang	52.998	0.0911	58.522

Figure 5. LSB technique for gray image

Figure 6. LSB technique for colour image

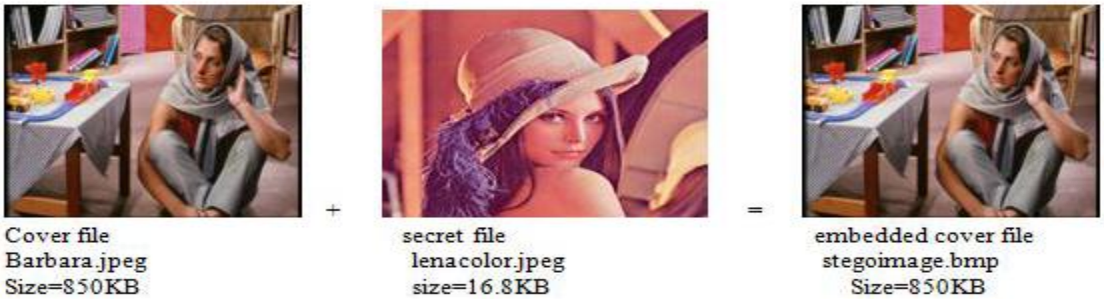


Figure 7. LSB Technique

4. Histogram-based Reversible Data Hiding

Reversible data hiding enables the embedding of messages in a host image without any loss of host content, which is proposed for image authentication that if the watermarked image is deemed authentic, we can revert it to the exact copy of the original image before the embedding occurred. In this paper, we present an improved histogram-based reversible data hiding scheme based on prediction and sorting. A rhombus prediction is employed to explore the prediction for histogram-based embedding. Sorting the prediction has a good influence on increasing the embedding capacity. Characteristics of the pixel difference are used to achieve large hiding capacity while keeping low distortion. The proposed scheme exploits a two-stage embedding strategy to solve the problem about communicating peak points. We also present a histogram shifting technique to prevent overflow and underflow. Performance comparisons with other existing reversible data hiding schemes are provided to demonstrate the superiority of the proposed scheme.

In [16], Ni *et al*. introduced a reversible data hiding scheme based on histogram modification that we will describe briefly in this section. The histogram modification technique involves generating histogram and finding the peak point and the zero point and shifting histogram bins to embed message bits. For a given host image, we first generate its histogram and find a peak point and a zero point. A peak point corresponds to the grayscale value which the maximum number of pixels in the given image assumes. On the contrary, a zero point corresponds to the grayscale value which no pixel in the given image assumes. For example, the histogram of the grayscale Lena image (512×512×8) is illustrated in figure 8, in which the peak point is at 154 and the zero point is at 255. Let P be the value of peak point and Z be the value of zero point. The range of the histogram, [P+1, Z-1], is shifted to the right-hand side by 1 to leave the zero point at P+1. Once a pixel with value P is encountered, if the message bit is “1,” increase the pixel value by 1. Otherwise, no modification is needed. We note that the number of message bits that can be embedded into an image equals to the number of pixels which are associated with the peak point.

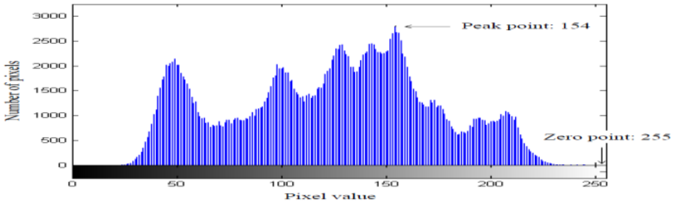


Figure 8. Histogram of the 'Lena' image

4.1. Rhombus Prediction and Sorting

Use of pixel difference histogram [17] introduced a significant performance advantage over previous methods. To improve our previous work, we present the prediction sorting to enhance the correlation of neighboring pixels. In order to predict the pixel value of position ui, j in Figure 9, we use a rhombus prediction by considering four neighboring pixels vi, j-1, vi-1, j, vi, j+1, vi+1, j. All pixels of the image are divided into two sets: the “White” set and “Gray” set. The pixel value u of the White set can be predicted by using the four neighboring pixel values of the Gray set and to hide data. Note that the two sets are independent, which means changes in one set do not affect the other set, and vice versa. The center pixel ui, j can be predicted from the four neighboring pixels vi, j-1, vi-1, j, vi, j+1, vi+1, j.

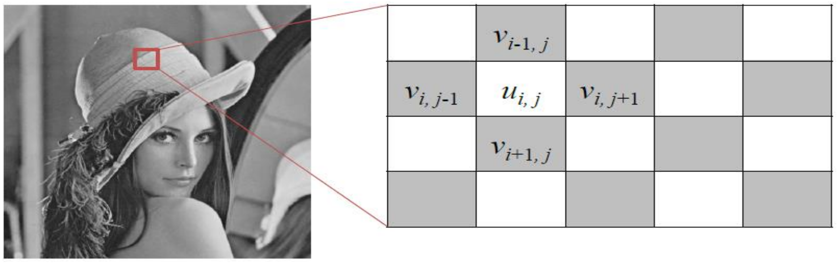


Figure 9. Rhombus prediction

4.2. Histogram Modification on Pixel Differences

The reversible data hiding scheme for White set is designed as follows.

1) Predict the pixel value ui, j in White set

2) Sort the host pixel ui, j according to the prediction value $u'i, j$, and produce the sorted pixels {$x0, x1,…, xi$} for $0 \leq i \leq N-1$ where N is the pixel number of White set.

3) Calculate the pixel difference di between pixels

4) Determine the peak point P from the pixel differences

5) If *di* > *P*, shift *xi* by 1 unit:

$$y_i = \begin{cases} x_i, & \text{if } i = 0 \text{ or } d_i < P, \\ x_i + 1, & \text{if } d_i > P \text{ and } x_i \geq x_{i-1}, \\ x_i - 1, & \text{if } d_i > P \text{ and } x_i < x_{i-1}, \end{cases} \quad \text{-- (1)}$$

Where *yi* is the watermarked value of pixel *i*.

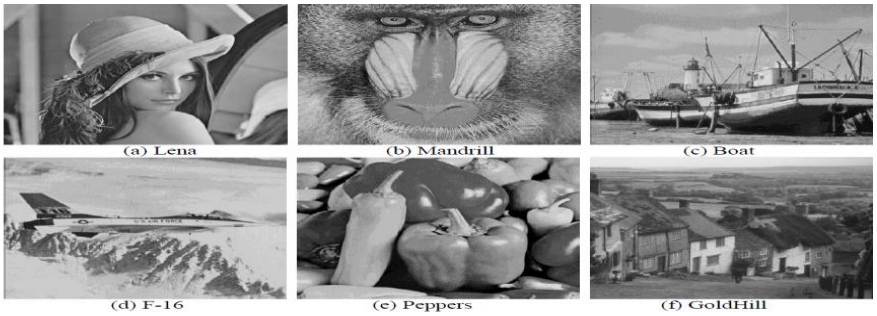
6) If *di* = *P*, modify *xi* according to the message bit:

$$y_i = \begin{cases} x_i + b, & \text{if } d_i = P \text{ and } x_i \geq x_{i-1}, \\ x_i - b, & \text{if } d_i = P \text{ and } x_i < x_{i-1}, \end{cases} \quad \text{-- (2)}$$

Where *yi* is the watermarked value of pixel *i*.

7) Construct the watermarked White set according to the sorted pixels {*y*0, *y*1, …, *yi*} for 0 ≤ *i* ≤ *N*-1 where *N* is the pixel number of White set.

Results are shown in following Figures 10 & Table 4.



(a) Lena (b) Mandrill (c) Boat

(d) F-16 (e) Peppers (f) GoldHill

Figure 10. Six test images used for performance evaluation

Table 4. Hiding capacity and distortion for six test images with *L*=1

Host image (512×512)	N_p	Pure payload *Cap* (bits)	Overhead information |*O*| (bits)	PSNR (dB)	Bit rate (bpp)
Lena	45074	45038	36	48.98	0.1718
Mandrill	14497	14361	136	48.38	0.0548
Boat	46461	46425	36	49.01	0.1771
F-16	65734	65698	36	49.54	0.2506
Peppers	31321	31285	36	48.69	0.1193
GoldHill	33537	33501	36	48.74	0.1278

5. RDH using optimum value Transfer

In this method [18] the optimal rule of value modification under a payload-distortion criterion is performed. By maximizing a target function using iterative algorithm, an optimal value transfer matrix can be obtained. Furthermore, we design a practical reversible data hiding scheme, in which the estimation errors of host pixels are used to accommodate the secret data and their values are modified according to the optimal value transfer matrix. This way, a good payload-distortion performance can be achieved.

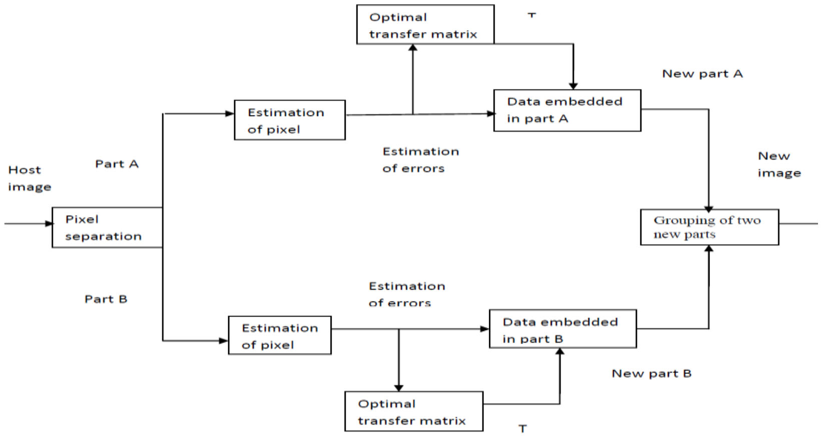


Figure 11. Data Embedding Procedure

5.1. Optimal Transfer Mechanism

In the RDH, using the optimal value transfer mechanism, it is a iterative procedure that is nothing but mathematical expression only. It will be in matrix format only; it will modify the cover values in RDH. In RDH optimal transfer matrix model is used.

To denote a histogram data H={…,h-3,h-2,h-1,h0,h1,h2,h3,…..}where hk= The amount of obtainable data with a value k. To denote the amount of available data possessing of original value i and new value j for data hiding as pi,j, and the transfer matrix is T= Px1,x2⋯Px1,x2⋮⋱⋮Px1,x2⋯Px1,x2 Where x1 and x2 are a minimum and maximum available cover data. In a optimal mechanism the difference Between two neighbouring pixel are doubled and the secret bit present in it is embedded. In the LSB the new difference value is q'=2.q+b here q ,q' and b are the original pixel-difference, new reference value and the secret bit ,here difference in the original value q will changed as 2q/(2q+1) where the secret bit b as 0/1.

5.2. Single-level-2D-DWT

In a single-level-2D-DWT, daubechisD4 transform is used; it is used for special cases applications, because it is used in finite data sets. It is a matrix technique in linear algebra in this scaling function and wavelet function is calculated. The daubechisD4 scaling and wavelet function coefficient can shift from right to left by two places in each iteration. There will be no overlapping occur .The scaling functions coefficients and wavelet transform are

DaubechiesD4 scaling functions:

ai = f0s2i+fis2i+1+f2s2i+2+f3s2i+3

a[i] = f0s[2i]+f1s[2i+1]+f2s[2i+2]+f3s[2i+3]

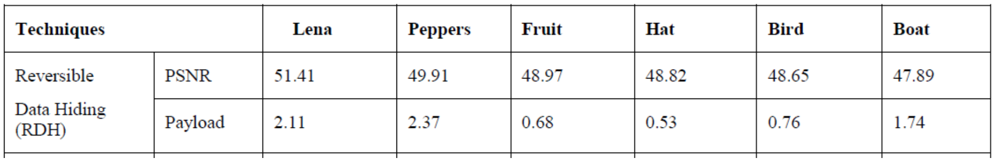
DaubechiesD4 wavelet function:

w = b0s2i+b1s2i+1+b2s2i+2+g3s2i+3

w[i] = b0s[2i]+b1s[2i+1]+b2s[2i+2]+b3s[2i+3]

Psnr and payload values are shown in following Table 5.

Table 5. PSNR and payload for various images using optimum value transfer



Techniques		Lena	Peppers	Fruit	Hat	Bird	Boat
Reversible Data Hiding (RDH)	PSNR	51.41	49.91	48.97	48.82	48.65	47.89
	Payload	2.11	2.37	0.68	0.53	0.76	1.74

6. Conclusion

In this paper, various reversible data hiding algorithms were discussed. Reversible data hiding algorithms plays a dominant role in security applications such as medical and military. All RDH techniques working better and give better results but they have drawbacks also. All the schemes need more improvement to enhance its performance. Furthermore, it is expected that more serious trials to improve the algorithms to hide data perfectly. All these algorithms mentioned in this paper are worked on data and images, there may be applicable to hide audio and video in future.

References

[1] W. Bender, D. Gruhl, N. Morimoto and A. Lu, “Techniques for data hiding”, IBM Systems Journal, vol. 35, **(1996)**, pp. 313-336,.

[2] C. W. Honsinger, P. W. Jones, M. Rabbani and J. C. Stoffel, “Lossless recovery of an original image containing embedded data”, U. S. Patent, Ed., **(2001)**.

[3] D. M. Thodi and J. J. Rodríguez, “Expansion Embedding Techniques for Reversible Watermarking”, IEEE Transactions on Image Processing, vol. 16, **(2007)**, pp. 721 – 730.

[4] C. D. Vleeschouwer, J. F. Delaigle and B. Macq, “Circular Interpretation of Histogram For reversible Watermarking”, In Proceedings of the IEEE 4th Workshop on Multimedia Signal Processing, **(2001)**, pp. 345–350.

[5] C. D. Vleeschouwer, J. F. Delaigle and B. Macq, “Circular interpretation of bijective transformations in lossless watermarking for media asset management”, IEEE Transactions on Multimedia, vol. 5, **(2003)**, pp. 97-105.

[6[J. -B. Feng, I. -C. Lin, C. -S. Tsai and Y. -P. Chu, “Reversible Watermarking: Current Status and Key Issues”, International Journal of Network Security vol. 2, **(2006)**, pp. 161–171.

[7] J. Fridrich, M. Goljan and R. Du, “Lossless Data Embedding For All Image Formats”, in proceedings of SIPE: Security and Watermarking of Multimedia Contents, **(2002)**, pp.572-583.

[8] M. U. Celik and A. M. Tekalp, “Lossless Generalized-LSB Data Embedding”, IEEE Transactions On Image Processing, vol. 14, **(2005)**, pp. 253-266.

[9] R. Anderson and F. Petitcolas, “On the limits of steganography”, IEEE Journal of Selected Areas in Communications, vol. 16, no. 4, May **(1998)**.

[10] A. Cheddad, J. Condell, K. Curran and P. M. Kevitt, “Digital image steganography: survey and analysis of current methods”, Signal Processing Journal, **(2010)**.

[11] J. R. Krenn, “Steganography and Steganalysis”, **(2004)** January.

[12] P. C. Mandal and B. P. Poddar, “Modern Steganographic technique: A Survey”, International Journal of Computer Science Engineering Technology (IJC- SET).

[13] M. Juneja, “Data hiding Algorithm for Bitmap Images using Steganography”, Department of computer science and Engineering, RBIEBT, Sahuran.

[14] Journal of Theoretical and Applied Information Technology, vol. 36, no.1, **(2012)** February.

[15] v. k. sharma and v. shrivastava, “A steganography algorithm for hiding image in Image by improved lsb substitution by minimize Detection”, Arya college of Engineering IT, Jaipur , Rajasthan (India).

[16] Z. Ni, Y. Q. Shi, N. Ansari and W. Su, “Reversible data hiding”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 3, **(2006)** March, pp. 354-362, Article (CrossRef Link).

[17] W. L. Tai, C. M. Yeh and C. C. Chang, “Reversible data hiding based on histogram modification of pixel differences”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 19, no. 6, **(2009)** June, pp. 906-910.

[18] X. Zhang, “Reversible Data Hiding With Optimal Value Transfer”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 15, no. 2, (2013), February, pp. 316-325.