VARIABLE RATE STEGANOGRAPHY IN DIGITAL IMAGES USING TWO, THREE AND FOUR NEIGHBOR PIXELS

Anita Pradhan
Department of CSE, Sri Sivani College of Engineering,
Srikakulam, Andhra Pradesh, India
anita.pradhan15@gmail.com

D.S. Sharma
Department of CSE, Sri Sivani College of Engineering,
Srikakulam, Andhra Pradesh, India
subhramanyasharma@gmail.com

Gandharba Swain
Department of IT, GMR Institute of Technology,
Rajam, Srikakulam, Andhra Pradesh, India,
gswain1234@gmail.com

Abstract

In this paper three image steganography methods with variable rate of embedding are proposed. In the first method, called two neighbor method the two neighboring pixels such as upper and left are exploited to take embedding decisions. In three neighbor method the three neighboring pixels such as upper, left and right are exploited to take embedding decisions. In four neighbor method the four neighboring pixels such as upper, left, right and bottom are exploited to take embedding decisions. The message extraction process is very simple and does not require any knowledge of the original image. The experimental results show that the distortion is less in two neighbor method as compared to the other two methods. But the theoretical studies reveal that the capacity in four neighbor method is more as compared to two and three neighbor methods.

Keywords: steganography, two neighbor method, three neighbor method, four neighbor method

1. Introduction

Information and communication technology has grown rapidly and internet is the most popular communication medium nowadays. But the message transmission over the internet is not secure. So we need secret communication methods for transmitting data over the internet. Cryptography is a well known method in which the information is encrypted by using a key and then sent on the channel. Steganography is yet another method in which the communication is not apparent to the intruder. It is an art of covert communication in which the secret information is hidden inside a carrier file such that the change in appearance of the carrier file is not apparent to normal human eye.

Steganography can be categorized into four categories. Those are: steganography in image, steganography in audio, steganography in video and steganography in text. When hiding information inside images usually Least Significant Bit (LSB) method is used. In the LSB method the 8^{th} bit of every byte of the carrier file is substituted by one bit of the secret information. This method works fine in the image carriers because if the least significant bit is changed from 0 to 1 or vice versa, there is hardly any change in the appearance of the color of that pixel. Instead of hiding a fixed number of bits in the LSBs of each pixel, one can also embed different number of bits in LSBs of different pixels based on pixel value range calculation [1].

In general if the pixels are located in edge areas they can tolerate larger changes than those in smooth areas. The range of changeable pixel value in smooth areas is small, where as in edge areas it is large so that the stego image maintains a good perceptual quality. Wu and Tsai proposed a pixel value differencing method, where a cover image is partitioned into non overlapping blocks of two consecutive pixels [2]. A difference value is calculated from the value of the two pixels in each block. Secret data is embedded into a cover image by replacing the difference values of the two pixel blocks of the cover image with similar ones, in which bits of embedded data are included. Zhang and Wang [3] found that pixel value differencing steganography is vulnerable to histogram based attacks and proposed a modification for enhanced security. Chang and Tseng employed two sided, three sided and four sided side match methods [4]. The two sided side match method uses

the side information of the upper and left neighboring pixels in order to make estimates. The three sided side match method uses upper and left neighboring pixels; and one of the other neighboring pixels. The four sided side match method uses all the upper, left, right and below neighbors. Kim et al. also proposed a method in which the fall off boundary problem (FOBP) is addressed [5]. Chang et al. proposed a three way pixel value differencing method [6]. Zhang et al. proposed a pixel value differencing technique by using the largest difference value among the other three pixels close to the target pixel to estimate how many secret bits will be embedded into the pixel [7].

This paper is comprised of four sections. In section2 our proposed two neighbor, three neighbor, and four neighbor methods are explained. In section3 the experimental results are discussed. In section4 the paper is concluded.

2. The proposed Methods

The proposed methods are discussed in the following sub sections.

2.1 Two neighbor Steganography

The cover image is scanned in the raster-scan order. The pixels in first row and first column are not embedded. The pixels are categorized into three categories as shown in fig.1. The pixels which are blue colored are used as target pixels, where data is to be hidden. The white colored pixels are used as neighbors. The pixels with cross marks are those which are neither treated as neighbor nor to hide data.

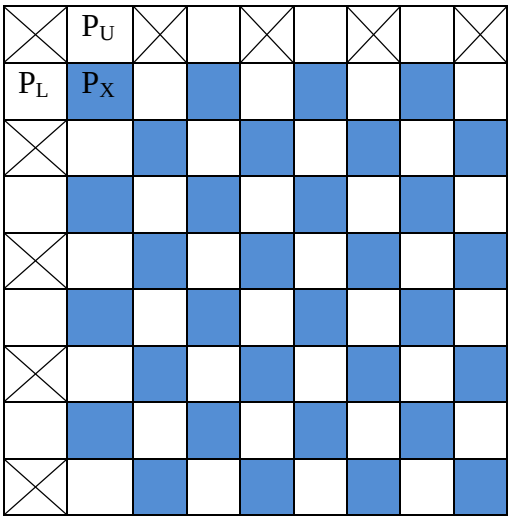


Fig.1. The target, neighbor and unused pixels in two neighbor method

The two neighbor method uses the upper and left neighboring pixels for estimating the number of bits to be embedded in the target pixel. Given an input pixel P_X with gray value g_x and; let g_u and g_l be the gray values of its upper pixel P_U and left pixel P_L respectively. Then a difference value d is computed as

$$d = g_{max} - g_{min} \quad (1)$$

where $g_{max} = \max(g_u, g_l)$ and $g_{min} = \min(g_u, g_l)$

The embedding capacity of the pixel depends on the value of d, let n be the number of bits which can be embedded in the target pixel P_X, then n is calculated by

$$n = \begin{cases} 1, & \text{if } 0 \le d \le 1 \\ \log_2 d, & \text{if } d > 1 \end{cases} \quad (2)$$

As the quality of the stego-image may be degraded drastically when n>4, we set n≡4. A sub-stream of n bits from the secret binary message is taken and is converted to integer b. Then the new value g_x' is computed as

$$g_x' = g_x - g_x \bmod 2^n + b \quad (3)$$

After replacing n-rightmost LSBs of g_x, we apply the following two adjustments shown in case1 and case2 below. Let δ_x be the difference between g_x and g_x' i.e. $\delta_x = g_x' - g_x$,

Case1: If $2^{n-1} < \delta_x < 2^n$ and $g_x' \ge 2^n$ then $g_x' = g_x' - 2^n$

Case 2: If $-2^n < \delta_x < -2^{n-1}$ and $g_x' < 256 - 2^n$ then $g_x' = g_x' + 2^n$.

The secret data is extracted from the stego-image by scanning it in raster scan order. Given a target pixel P_X^* with gray value g_x^*; let g_u^*, g_l^* be the gray values of its upper neighboring pixel P_U^* and left neighboring pixel P_L^* respectively. Then gray value difference d^* is defined as

$$d^* = g_{max}^* - g_{min}^* \quad (4)$$

where $g_{max}^* = \max(g_u^*, g_l^*)$ and $g_{min}^* = \min(g_u^*, g_l^*)$

Let n^* be the number of bits which can be extracted from the input pixel P_X^*. The value n^* is calculated by

$$n^* = \begin{cases} 1, & \text{if } 0 \le d^* \le 1 \\ \log_2 d^*, & \text{if } d^* > 1 \end{cases} \quad (5)$$

if $n^* > 4$, we set $n \equiv 4$ and the value b is calculated by

$$b = g_x^* \bmod 2^{n^*} \quad (6)$$

Finally, n^* bits secret data can be obtained by converting the value b to a binary string.

2.2 Three neighbor Steganography

The cover image is scanned in the raster-scan order. The first row, first column and last column are not embedded. The pixels are categorized into three categories as shown in fig.2. The pixels which are blue colored are used as target pixels, where data is to be hidden. The white colored pixels are used as neighbors. The pixels with cross marks are those which are neither treated as neighbor nor to hide data.

The three neighbor method uses the upper, left and right neighboring pixels for estimating the number of bits to be embedded in the target pixel. Let g_x be the gray value of pixel P_X ; g_u, g_l and g_r be the gray values of its upper neighboring pixel P_U, left neighboring pixel P_L and right neighboring pixel P_R respectively. Then the difference value d is computed as

$$d = g_{max} - g_{min} \quad (7)$$

where $g_{max} = \max(g_u, g_l, g_r)$ and $g_{min} = \min(g_u, g_l, g_r)$

The embedding capacity of pixel depends on the value of d, let n be the number of bits which can be embedded in the target pixel P_X, then n is calculated by

$$n = \begin{cases} 1, & \text{if } 0 \le d \le 1 \\ \log_2 d, & \text{if } d > 1 \end{cases} \quad (8)$$

As the image quality of the stego-image may be degraded drastically when n>4, we set $n \equiv 4$. A sub-stream of n bits from the secret binary message is taken and is converted to integer b. Then the new value g_x' is computed as

$$g_x' = g_x - g_x \bmod 2^n + b \quad (9)$$

After replacing n-rightmost LSBs of g_x, we apply the following two adjustments shown in case1 and case2 below. Let δ_x be the difference between g_x and g_x' i.e. $\delta_x = g_x' - g_x$,

Case1: If $2^{n-1} < \delta_x < 2^n$ and $g_x' \ge 2^n$ then $g_x' = g_x' - 2^n$

Case 2: If $-2^n < \delta_x < -2^{n-1}$ and $g_x' < 256 - 2^n$ then $g_x' = g_x' + 2^n$

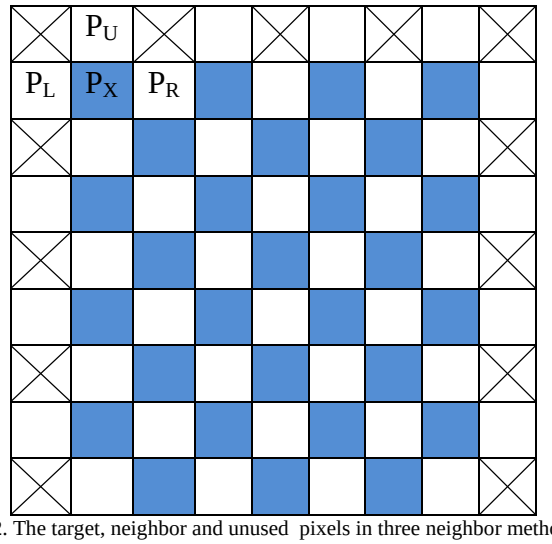


Fig.2. The target, neighbor and unused pixels in three neighbor method

The secret data is extracted from the stego-image by scanning it in raster-scan order. Given a target pixel P_X^* with gray value g_x^* ; let g_u^*, g_l^* and g_r^* be the gray values of its upper neighboring pixel P_U^*, left neighboring pixel P_L^* and right neighboring pixel P_R^* respectively. The gray value difference d^* is defined as

$$d^* = g_{max}^* - g_{min}^* \quad (10)$$

where $g_{max}^* = \max(g_u^*, g_l^*, g_r^*)$ and $g_{min}^* = \min(g_u^*, g_l^*, g_r^*)$

Let n^* be the number of bits which can be extracted from the input pixel P_X^*. The value n^* is calculated by

$$n^* = \begin{cases} 1, & \text{if } 0 \le d^* \le 1 \\ \log_2 d^*, & \text{if } d^* > 1 \end{cases} \quad (11)$$

if $n^* > 4$, we set $n \equiv 4$ and the value b is calculated by

$$b = g_x^* \bmod 2^{n^*} \quad (12)$$

Finally, n^* bits secret data can be obtained by converting the value b to a binary string.

2.3 Four neighbor Steganography

The cover image is scanned in the raster-scan order. The pixels are categorized into three categories as shown in fig.3. The pixels which are blue colored are used as target pixels, where data is to be hidden. The white colored pixels are used as neighbors. The pixels with cross marks are those which are neither treated as neighbor nor to hide data.

The four sided side match uses the upper, left, right and bottom neighboring pixels for estimating the number of bits to be embedded. Let g_x be the gray value of pixel P_X ; g_u, g_l , g_r and g_b be the gray values of its upper pixel P_U, left pixel P_L , right pixel P_R and bottom pixel P_B respectively. Then the difference value d is computed as

$$d = g_{max} - g_{min} \quad (13)$$

where $g_{max} = \max(g_u, g_l, g_r, g_b)$ and $g_{min} = \min(g_u, g_l, g_r, g_b)$

The embedding capacity of pixel depends on the value of d, let n be the number of bits which can be embedded in the target pixel P_X, then n is calculated by

$$n = \begin{cases} 1, & \text{if } 0 \le d \le 1 \\ \log_2 d, & \text{if } d > 1 \end{cases} \quad (14)$$

As the image quality of the stego-image may be degraded drastically when n>4, we set n≡4. A sub-stream of n bits from the secret binary message is taken and is converted to integer b. Then the new value g_x' is computed as

$$g_x' = g_x - g_x \bmod 2^n + b \quad (15)$$

After replacing n-rightmost LSBs of g_x, we apply the following two adjustments shown in case1 and case2 below. Let δ_x be the difference between g_x and g_x' i.e. $\delta_x = g_x' - g_x$,

Case1: If $2^{n-1} < \delta_x < 2^n$ and $g_x' \ge 2^n$ then $g_x' = g_x' - 2^n$

Case 2: If $-2^n < \delta_x < -2^{n-1}$ and $g_x' < 256 - 2^n$ then $g_x' = g_x' + 2^n$

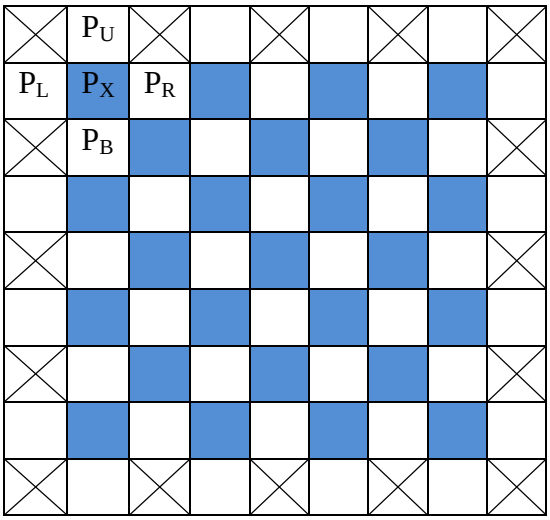


Fig.3. The target, neighbor and unused pixels in four neighbor method

The secret data is extracted from the stego-image by scanning it in raster-scan order. Given a target pixel P_X^* with gray value g_x^*; let g_u^*, g_l^*, g_r^* and g_b^* be the gray values of its upper neighboring pixel P_U^*, left neighboring pixel P_L^*, right neighboring pixel P_R^* and bottom neighboring pixel P_B^* respectively. The gray value difference d^* is defined as

$$d^* = g_{max}^* - g_{min}^* \qquad (16)$$

where $g_{max}^* = \max(g_u^*, g_l^*, g_r^*, g_b^*)$ and $g_{min}^* = \min(g_u^*, g_l^*, g_r^*, g_b^*)$

Let n^* be the number of bits which can be extracted from the input pixel P_X^*. The value n^* is calculated by

$$n^* = \begin{cases} 1, & \text{if } 0 \le d^* \le 1 \\ \log_2 d^*, & \text{if } d^* > 1 \end{cases} \qquad (17)$$

if $n^* > 4$, we set $n \equiv 4$ and the value b is calculated by

$$b = g_x^* \bmod 2^{n^*} \qquad (18)$$

Finally, n^* bits secret data can be obtained by converting the value b to a binary string.

3. Experimental Results and Discussion

3.1 Experimental results

The techniques are implemented using Matlab and are tested with many images. The observations for three standard images are shown. Fig.4.(a) is the Airplane image with a size of 192 kilo bytes. Fig.4.(b), (c), (d) are the stego images in two neighbor, three neighbor and four neighbor methods with 2048 bytes of data hidden in each and (e), (f), (g) and (h) are their histograms. Fig.5 is the Boat image with a size of 768 kilo bytes. Fig.5.(b), (c), (d) are the stego images in two neighbor, three neighbor and four neighbor methods with 1024 bytes of data hidden in each and (e), (f), (g) and (h) are their histograms. Fig.6 is the House image with a size of 768 kilo bytes. Fig.6.(b), (c), (d) are the stego images in two neighbor, three neighbor and four neighbor methods with 1024 bytes of data hidden in each and (e), (f), (g) and (h) are their histograms.

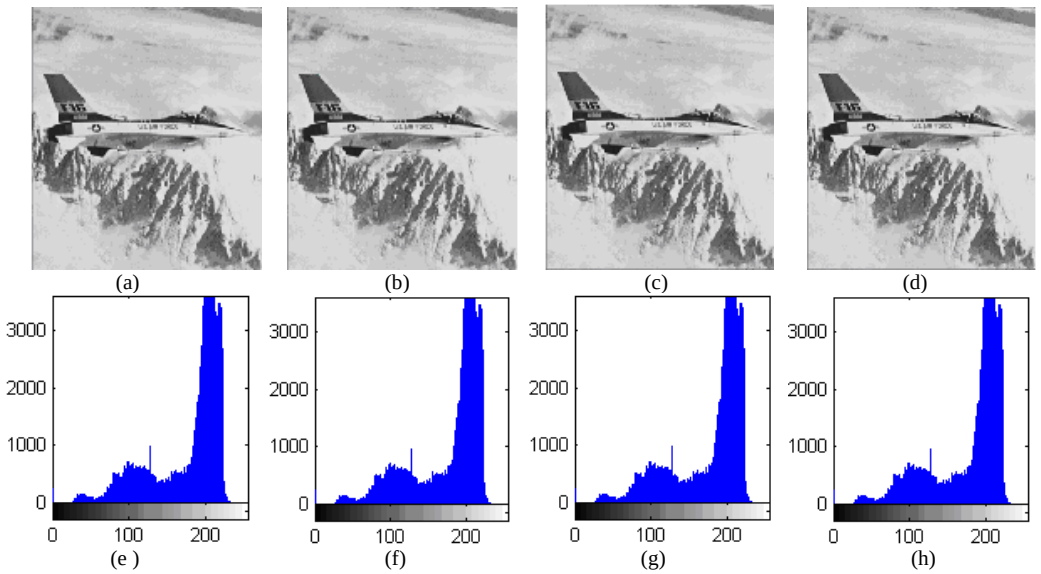
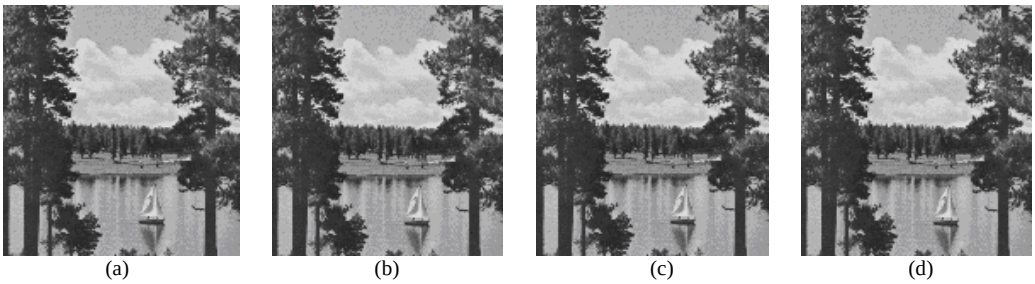


Fig.4. (a) Original Airplane image, (b), (c) and (d) are stego images, (e), (f), (g) and (h) are their histograms



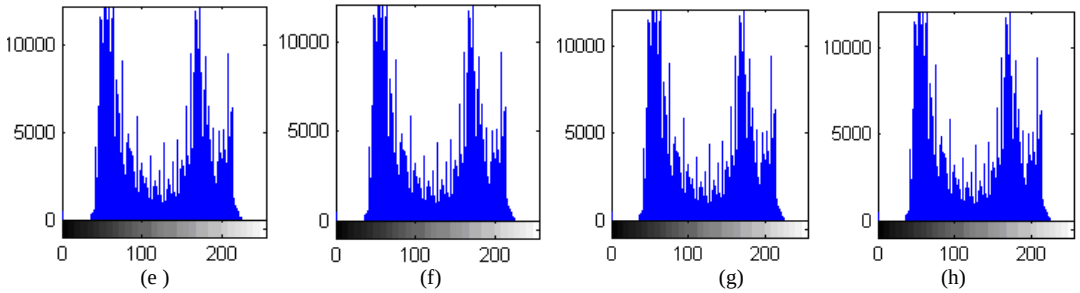


Fig.5. (a) Original Boat image, (b), (c) and (d) are stego images, (e), (f), (g) and (h) are their histograms

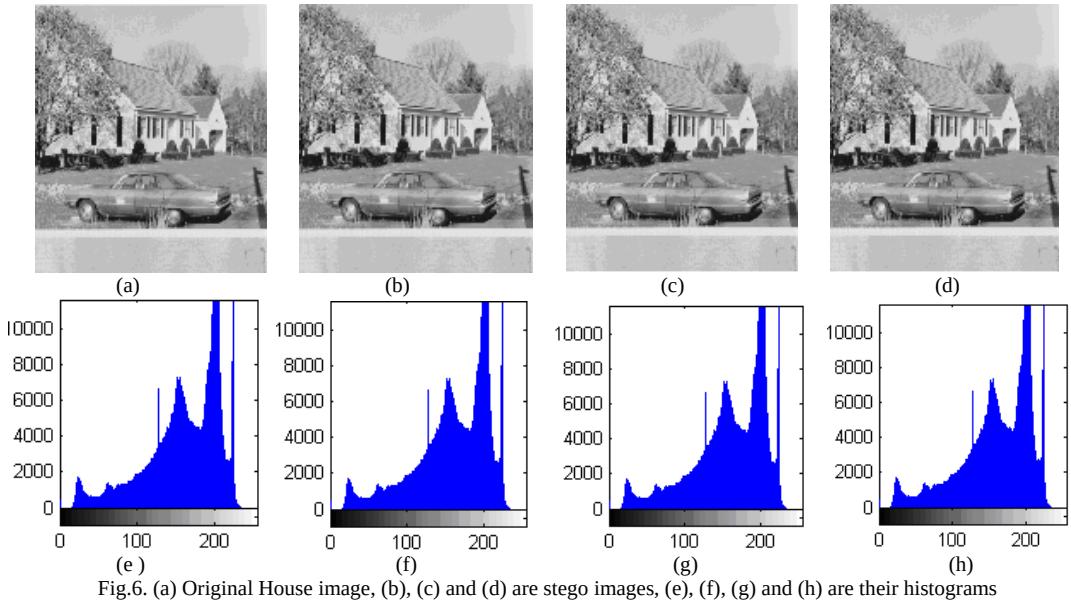


Fig.6. (a) Original House image, (b), (c) and (d) are stego images, (e), (f), (g) and (h) are their histograms

3.2 Discussion

It is seen from these figures that the distortions resulted from embedding are imperceptible to human vision. It means that such distortions are not noticeable because changes in edge areas of images are generally less conspicuous to human eyes. The amount of hidden data with peak signal to noise ratio (PSNR) values are shown in table-1. The lesser is the PSNR means more is the distortion. With same amount of hidden data the distortion is minimum in two neighbor case than compared to three and four neighbor cases. As per the observations; the four neighbor case is always getting the minimum PSNR. But four neighbor method can hide more amount of data compared to two and three neighbor methods. This is because the gray value difference d computed (in equations 1, 7 and 13) is largest in four neighbor method.

The performance of various steganographic methods can be rated by the three parameters: security, capacity, and imperceptibility. The steganographic methods proposed in this paper are very secure as variable number of bits are hidden in different target pixels. Capacity means the amount of message that can be embedded. To be useful in conveying secret message, the hiding capacity provided by steganography should be as high as possible, which may be given in absolute measurement such as the size of secret message, or in relative value called data embedding rate, such as bits per pixel, or the ratio of the secret message to the cover medium. We can see in fig.1, fig.2 and fig.3 that more than 25% of the pixels can be used to hide data and in each pixel up to a maximum of 4 bits. Thus the capacity is also good. Under the same level of security and capacity, the higher the imperceptibility of the stego-image, the better is the steganography method. If the resultant stego-image appears innocuous enough, one can believe this requirement to be satisfied well. In fig.4, fig.5 and fig.6 it can be observed that the stego images are looking very innocuous. The distortions in quality of the stego images are not noticeable to human vision.

Instead of replacing fixed number of LSBs in different pixels directly, the proposed methods replace variable number of bits in different pixels. Thus the proposed methods are more secure than LSB method. To make the methods two fold secure we can encrypt the secret data by data encryption standard (DES) algorithm and then embed the cipher data into the image.

Table-1. Experimental results for the three methods

Image Name	Image Size (in kilo bytes)	Amount of hidden data (in bytes)	Peak Signal to Noise Ratio (decibels)		
			Two neighbor	Three neighbor	Four neighbor
Airplane	192	2048	52.31	52.33	49.93
Boat	768	1024	51.83	51.38	50.94
House	768	1024	55.88	55.04	54.83

4. Conclusion

In this paper three image steganography methods with variable rate of embedding is proposed. The embedding capacity and peak signal to noise ratio (PSNR) values are acceptable in all the three cases. With same amount of hidden data the distortion is minimum in two neighbor case than compared to three and four neighbor cases. As per the observations shown in table-1; the four neighbor method is getting the minimum PSNR. But four neighbor method can hide more amount of data compared to two and three neighbor methods. This is because the gray value difference d computed (in equations 1, 7 and 13) is largest in four neighbor method. After the information is embedded the change in quality of the images are not apparent.

References

[1] Jain, Y. K.; Ahirwal, R. R. (2010): A Novel Image Steganography Method with Adaptive Number of Least Significant Bits Modification Based on Private Stego Keys, International Journal of Computer Science and Security, vol.4, no.1, pp.40-49.

[2] Wu, D. C.; Tsai, W. H. (2003): A Steganograhic Method for Images by Pixel Value Differencing, Pattern Recognition Letters, vol.24, no.9-10, pp.1613-1626.

[3] Zhang, X.; Wang, S. (2004): Vulnerability of Pixel Value Differencing Steganography to Histogram Analysis and Modification for Enhanced Security, Pattern Recognition Letters, vol.25, pp. 331-339.

[4] Chang, C. C.; Tseng, H. W. (2004): A Steganographic Method for Digital Images Using Side match, Pattern Recognition Letters, vol.25, no.12, pp.1431-1437.

[5] Kim, K. J.; Jung, K. H.; Yoo, K.Y. (2008): Image Steganographic Method with Variable Embedding Length, International Symposium on Ubiquitous Computing, pp. 210-213.

[6] Chang, K.C.; Chang, C. P.; Huang, P. S.; Tu, T.M. (2008): A Novel Image Steganography method Using Tri-way Pixel Value Differencing, Journal of Multimedia, vol.3, no.2, pp.37-44.

[7] Zhang, H. L.; Geng, G. Z.; Xiong, C. Q. (2009): Image Steganography using Pixel-Value Differencing, Second International Conference on Electronic Commerce and Security, pp.109-112.