



# Steganography using two sided, three sided, and four sided side match methods

Gandharba Swain · Saroj Kumar Lenka

Received: 13 May 2012 / Accepted: 17 April 2013 / Published online: 7 May 2013  
© CSI Publications 2013

**Abstract** In this paper the two sided, three sided and four sided side match steganography methods are proposed. In two sided side match method the two neighboring pixels such as upper and upper-right corner are exploited to take embedding decision. In three sided side match method the three neighboring pixels such as upper, upper-right corner and upper-left corner are exploited to take embedding decision. In four sided side match method the four neighboring pixels such as upper, left, upper-right corner and upper-left corner are exploited to take embedding decision. In each of these methods the fall off boundary problem (FOBP) and fall in error problem (FIEP) are addressed. The FOBP and FIEP conditions are checked while embedding at the sender and while extracting at the receiver. These methods possess large hiding capacity, better imperceptibility and high security.

**Keywords** Steganography · Side match · Pixel value differencing · Fall off boundary problem · Fall in error problem

## 1 Introduction

Information and communication technology has grown rapidly and internet is the most popular communication

medium now-a-days. But message transmission over the internet is facing some problems such as data security, copyright control, etc. So we need secret communication methods to transmit 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 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 noticeable by normal human eye. Sometimes we use both cryptography and steganography to achieve two levels of security.

Steganography can be categorized into three categories [1]. Those are: steganography in image, steganography in audio, and steganography in video. In recent past steganography in text has been evolved. The image steganography techniques can be categorized into two types such as spatial domain and frequency domain. In each of these categories we can have adaptive and dynamic methods. Adaptive methods are image statistics based, where as dynamic methods are message bit dependent.

When hiding information inside images usually least significant bit (LSB) substitution method is used. In the LSB substitution method the 8th bit (LSB 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, the pixel value may change hardly by  $\pm 1$ . Instead of hiding a fixed number of bits in the LSBs of each pixel, one can also hide different number of bits in LSBs of different pixels based on pixel value range calculation [2]. In image steganography, a pixel can carry secret bits by adding 1 to the pixel gray value or subtracting 1 from the pixel gray value. This kind of  $\pm 1$  steganography can hide a

---

G. Swain (✉)  
Department of CSE, SOA University, Bhubaneswar 751030,  
Odisha, India  
e-mail: gswain1234@gmail.com

S. K. Lenka  
Department of CSE, MITS University, Lakshmanagarh 332311,  
Rajasthan, India  
e-mail: lenka.sarojkumar@gmail.com

longer message than simple LSB embedding. Zhang et al. [3] proposed a double layered embedding method to further improve the embedding efficiency of  $\pm 1$  steganography.

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 [4] proposed a pixel value differencing (PVD) method, where a cover image is partitioned into non-overlapping blocks of two consecutive pixels. 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 [5] found that pixel value differencing steganography is vulnerable to histogram based attacks and proposed a modification for enhanced security. Chang et al. [6] proposed a three way pixel value differencing method by considering three directional differences in each block of four pixels. Chang and Tseng [7] proposed two sided, three sided and four sided side match methods. The two sided side match method uses the two neighboring pixels such as upper and left to estimate the number of bits to be embedded in a target pixel. The three sided side match method is three types. In type 1, the three neighboring pixels such as upper, left and right are used in estimation. In type 2, the three neighboring pixels such as upper, left and below are used in estimation and in type 3, the four corner neighbors are used in estimation. The four sided side match method uses four neighboring pixels such as upper, left, right and below to take embedding decisions. Kim et al. [8] also proposed a method using side match in which the fall off boundary problem (FOBP) is addressed. Zhang et al. [9] proposed a pixel value differencing method by using the largest difference value among the other three pixels close to the target pixel to estimate the number of bits that will be embedded into the target pixel. But a technical problem called as fall in error problem (FIEP) is not addressed in these methods. In our proposed two sided, three sided and four sided side match methods both FOBP and FIEP problems are taken care.

This paper is comprised of four sections. In Sect. 2 our proposed methods are explained. In Sect. 3 the experimental results are discussed and in Sect. 4 the paper is concluded.

## 2 The proposed methods

The two sided, three sided and four sided side match methods proposed in [7] do not address the FIEP. The two sided, three sided and four sided side match methods proposed in this paper addresses both FIEP and FOBP problems. These are discussed in the following sub sections.

### 2.1 Two sided side match steganography

The cover image is scanned in the raster-scan order. The pixels are categorized into three categories as shown in Fig. 1. The pixels shaded with gray color are used for embedding and as neighbors. The white colored pixels are used as neighbors only. The pixels shaded with light gray color are neither used for embedding nor as neighbors.

The two sided side match method uses the upper and upper-right neighbor pixels for estimating the number of bits to be embedded in a target pixel. Suppose  $P_x$  is a target pixel with gray value  $g_x$ ,  $P_U$  is its upper neighboring pixel with gray value  $g_u$  and  $P_{UR}$  is its upper-right neighboring pixel with gray value  $g_{ur}$ . Then to estimate the number of bits that can be embedded in the target pixel, a difference value  $d$  is computed as in Eq. 1.

$$d = (g_u + g_{ur})/2 - g_x \tag{1}$$

If the pixels are in a smooth area we will get a smaller value of  $d$  and if the pixels are in an edge area we will get a larger value of  $d$ . Larger the value of  $d$  means more number of bits can be embedded. If  $d$  has a value  $-1, 0$ , or  $1$  then the pixel is not used for embedding. If  $d$  has a value  $\geq 2$  then we can hide  $n$  bits in that pixel. The value  $n$  is calculated as in Eq. 2.

$$n = \log_2|d|, \quad \text{if } |d| > 1 \tag{2}$$

Now a sub-stream of  $n$  bits of secret data is taken, converted to integer value  $b$  and the new difference value  $d'$  is computed as in Eq. 3.

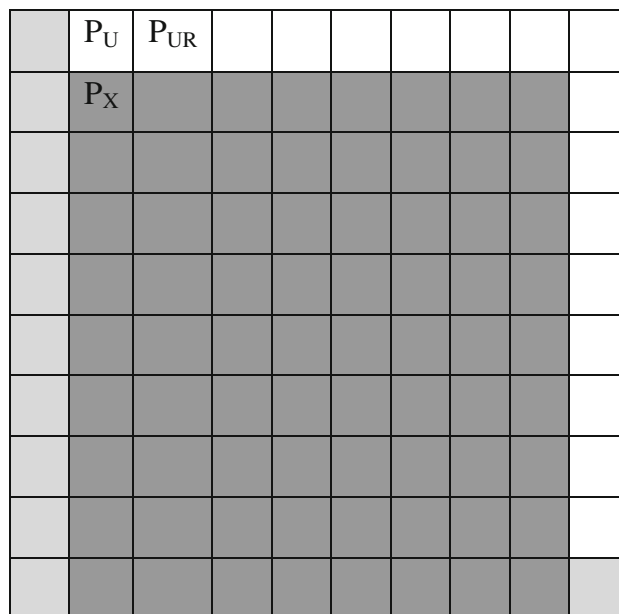


Fig. 1 Types of pixels in two sided side match method

$$d' = \begin{cases} 2^n + b, & \text{if } d > 1; \\ -(2^n + b), & \text{if } d < 1. \end{cases} \quad (3)$$

Hence the new value of the pixel after embedding  $n$  secret bits is computed as in Eq. 4.

$$g'_x = (g_u + g_{ur})/2 - d' \quad (4)$$

The extraction procedure is also very simple. The embedded data is extracted in raster-scan order. Suppose  $P_x^*$  is the target pixel with gray value  $g_x^*$ ,  $P_U^*$  is its upper neighboring pixel with gray value  $g_u^*$  and  $P_{ur}^*$  is its upper-right neighboring pixel with gray value  $g_{ur}^*$ . Then to estimate the number of bits embedded, a difference value  $d^*$  is computed as in Eq. 5.

$$d^* = (g_u^* + g_{ur}^*)/2 - g_x^* \quad (5)$$

If  $d^*$  has a value  $-1, 0$  or  $1$ , then that pixel is ignored. Otherwise the number of bits, say  $n$ , which was embedded, is calculated by Eq. 6.

$$n = \log_2 |d^*|, \quad \text{if } |d^*| > 1 \quad (6)$$

Now the integer equivalent of embedded bit stream,  $b$  is computed as in Eq. 7.

$$b = \begin{cases} d^* - 2^n, & \text{if } d^* > 1; \\ -d^* - 2^n, & \text{if } d^* < 1. \end{cases} \quad (7)$$

This integer value  $b$  is then converted to  $n$  bit binary.

Sometimes, the new value of pixel  $P_x$  may fall off boundary of the range  $\{0, 255\}$ , in the following cases as discussed in [7].

Case 1:  $d > 1$  and  $(g_u + g_{ur})/2 < 2^{n+1} - 1$ .

From Eqs. 3 and 4,  $g'_x = (g_u + g_{ur})/2 - d' = (g_u + g_{ur})/2 - (2^n + b)$ .

Assume that the embedding value  $b$  is given by the largest value  $2^n - 1$ , then.

$$g'_x = (g_u + g_{ur})/2 - (2^n + 2^n - 1) = (g_u + g_{ur})/2 - (2^{n+1} - 1).$$

Therefore,  $g_x$  will be a negative quantity when  $(g_u + g_{ur})/2 < (2^{n+1} - 1)$ .

Case 2:  $d < 1$  and  $(g_u + g_{ur})/2 + 2^{n+1} > 256$ .

From Equations 3 and 4,  $g'_x = (g_u + g_{ur})/2 - d' = (g_u + g_{ur})/2 + (2^n + b)$ . Assume that the embedding value  $b$  is given by the largest value  $2^n - 1$ , then.

$$g'_x = (g_u + g_{ur})/2 + (2^n + 2^n - 1) = (g_u + g_{ur})/2 + 2^{n+1} - 1.$$

Therefore,  $g'_x$  will be  $>255$  when  $(g_u + g_{ur})/2 + 2^{n+1} > 256$ .

Suppose for  $d = -1, 0$  or  $1$  only one bit data will be embedded using traditional LSB substitution method as it is done by Chang and Tseng [7]. An error condition occurs for  $d = -1$ , the extraction at receiver fails. Let us see an example. Suppose in two neighbor case  $g_x = 32, g_u = 31$

and  $g_{ur} = 31$ . Then  $d = (31 + 31)/2 - 32 = -1$ . The value of  $g_x = 32$  in binary it is 00100000. For  $d = -1$ , only one bit data is to be embedded by using LSB method. Suppose the one bit to be embedded is bit 1, then the new value of  $g_x$  after embedding is 00100001 = 33. Now at receiver  $g_x^* = 33, g_u^* = 31$  and  $g_{ur}^* = 31, d^* = (31 + 31)/2 - 33 = -2$ . As  $d^*$  at receiver is not 1, 0 or  $-1$ , so by applying Eq. 6 we get,  $n = \log_2 |d^*| = 1$ , Next by using Eq. 7, we get  $b = -d^* - 2^1 = 2 - 2 = 0$ , the bit extracted is 0, but the embedded bit is 1. Thus this is an error condition. It is better to include the conditions  $d = 0$  and  $d = 1$  too for the sake of uniformity. So a pixel where  $d = -1, 0$  or  $1$  is said to FIEP. This is to be considered as case 3.

Case 3: If  $d = -1, 0$  or  $1$  then do not embed in that pixel.

The falling off boundary checking and the falling in error checking is applied at the sender for data embedding and at the receiver for data extraction. The pixel  $P_x$  which falls in these three conditions is not embedded at the sender. Similarly data is not extracted from it at the receiver.

Let us discuss an example for embedding and extraction. Suppose the pixel  $P_x$  has a gray value  $g_x = 45$ , its upper neighbor pixel  $P_U$  has a gray value  $g_u = 40$  and its upper-right neighbor pixel  $P_{UR}$  has a gray value  $g_{ur} = 42$ . Then  $d = (40 + 42)/2 - 45 = -4$  as per Eq. 1. The number of bits  $n$  that can be embedded in pixel  $P_x$  is 2 as per Eq. 2. Suppose the two binary bits to be embedded are  $(10)_2$ . It's integer value is 2, i.e.  $b = 2$ . Then  $d' = -6$  as per Eq. 3 and the new value of pixel after embedding,  $g'_x = (g_u + g_{ur})/2 - d' = 41 - (-6) = 47$ .

At receiver the pixel  $P_x^*$  has gray value  $g_x^* = 47$ , it's upper neighbor  $P_U^*$  has gray value,  $g_u^* = 40$  and upper-right neighbor  $P_{UR}^*$  has gray value,  $g_{ur}^* = 42$ . Then the difference  $d^* = (g_u^* + g_{ur}^*)/2 - g_x^* = 41 - 47 = -6$  as per Eq. 5. The number of bits that is to be extracted is,  $n = \log_2 |-6| = 2$  (the lower rounded integer value is taken). Thus  $b = 2$  as per Eq. 7. In two bit binary the integer 2 is represented as  $(10)_2$ , which are the two bits embedded at the sender.

### 2.2 Three sided side match steganography

The cover image is scanned in the raster-scan order. The pixels are categorized into three categories as shown in Fig. 2. The pixels shaded with gray color are used for embedding and as neighbors. The white colored pixels are used as neighbors only. The pixels shaded with light gray color are neither used for embedding nor as neighbors.

The three sided side match method uses the upper, upper-right and upper-left neighbor pixels for estimating the number of bits to be embedded in a target pixel. Suppose  $P_x$  is a target pixel with gray value  $g_x, P_U$  is its upper neighboring pixel with gray value  $g_u, P_{UR}$  is its upper-right

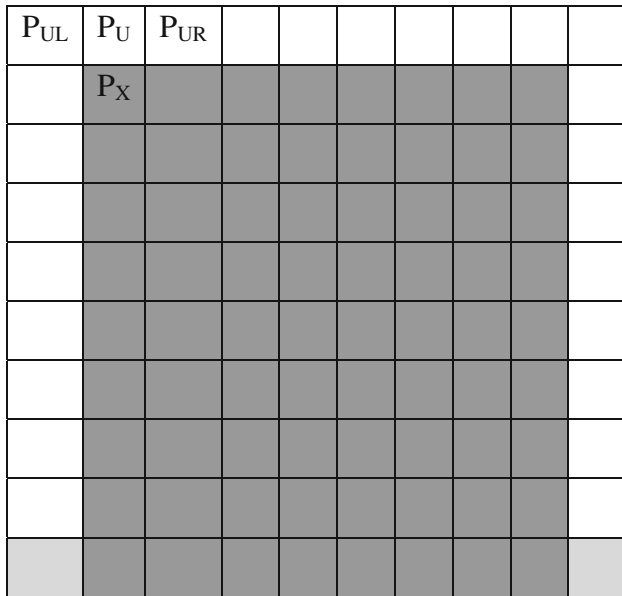


Fig. 2 Types of pixels in three sided side match method

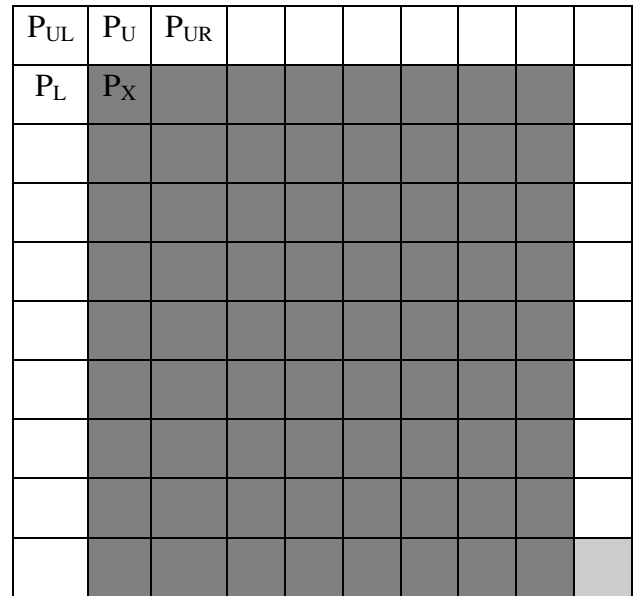


Fig. 3 Types of pixels in four sided side match method

neighboring pixel with gray value  $g_{ur}$ , and  $P_{UL}$  is its upper-left neighboring pixel with gray value  $g_{ul}$ . Then to estimate the number of bits to be embedded, a difference value  $d$  is computed as in Eq. 8.

$$d = (g_u + g_{ur} + g_{ul})/3 - g_x \tag{8}$$

If  $d$  has a value  $-1, 0$ , or  $1$  then in that pixel no data is embedded. If  $d$  has a value  $\geq 2$  then we can hide  $n$  bits in that pixel. The value  $n$  is calculated as in Eq. 9.

$$n = \log_2|d|, \quad \text{if } |d| > 1 \tag{9}$$

Now a sub-stream of  $n$  bits of secret data is taken, converted to integer value  $b$  and the new difference value  $d'$  is computed as in Eq. 10.

$$d' = \begin{cases} 2^n + b, & \text{if } d > 1; \\ -(2^n + b), & \text{if } d < 1. \end{cases} \tag{10}$$

Hence the new value of the pixel after embedding  $n$  secret bits is computed by Eq. 11.

$$g'_x = (g_u + g_{ur} + g_{ul})/3 - d' \tag{11}$$

The extraction procedure is identical to that of two sided side match method. The FOBP and FIEP conditions are also taken care as in two sided side match method.

### 2.3 Four sided side match 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 shaded with gray color are used for

embedding and as neighbors. The white colored pixels are used as neighbors only. The pixel shaded with light gray color is neither used for embedding nor as neighbor.

The four sided side match method uses the upper, upper-right, upper-left and left neighbor pixels for estimating the number of bits to be embedded in a target pixel. Suppose  $P_X$  is a target pixel with gray value  $g_x$ ,  $P_U$  is its upper neighboring pixel with gray value  $g_u$ ,  $P_{UR}$  is its upper-right neighboring pixel with gray value  $g_{ur}$ ,  $P_{UL}$  is its upper-left neighboring pixel with gray value  $g_{ul}$  and  $P_L$  is its left neighboring pixel with gray value  $g_l$ . Then to estimate the number of bits to be embedded, a difference value  $d$  is computed as in Eq. 12.

$$d = (g_u + g_{ur} + g_{ul} + g_l)/4 - g_x \tag{12}$$

If  $d$  has a value  $-1, 0$ , or  $1$  then in that pixel no data is embedded. If  $d$  has a value  $\geq 2$  then we can hide  $n$  bits in that pixel. The value  $n$  is calculated by Eq. 13.

$$n = \log_2|d|, \quad \text{if } |d| > 1 \tag{13}$$

Now a sub-stream of  $n$  bits of secret data is taken, converted to integer value  $b$  and the new difference value  $d'$  is computed as in Eq. 14.

$$d' = \begin{cases} 2^n + b, & \text{if } d > 1; \\ -(2^n + b), & \text{if } d < 1. \end{cases} \tag{14}$$

Hence the new value of the pixel after embedding  $n$  secret bits is computed by Eq. 15.

$$g'_x = (g_u + g_{ur} + g_{ul} + g_l)/4 - d' \tag{15}$$

The extraction procedure is identical to that of two sided side match method. The FOBP and FIEP conditions are also taken care as in two sided side match method.

### 3 Experimental results and discussion

#### 3.1 Experimental results

The methods are implemented using MATLAB and are tested with many images. The observations for three standard images are shown and discussed. Figure 4a is the Airplane image with a size of 192 kb. Figure 4b–d are the stego-images in two, three and four sided side match methods respectively with 10,240 bytes of data hidden in each. Figure 4e–h are their histograms. Figure 5 is the Lena image with a size of 535 kb. Figure 5b–d are the stego-images in two, three and four sided side match methods with 20,480 bytes of data hidden in each. Figure 5e–h are their histograms. Figure 6 is the Baboon image with a size of 525 kb. Figure 6b–d are the stego-images in two, three and four sided side match methods respectively with 12,288 bytes of data hidden in each. Figure 6e–h are their histograms.

#### 3.2 Discussion

The performance of various steganographic methods can be rated by the three parameters: security, capacity, and imperceptibility [10]. The steganographic methods

proposed in this paper are secure because instead of replacing the LSBs of pixel values directly, the proposed methods adaptively change the pixel value into another value based on the values of the neighboring pixels. The traditional LSB steganography have a common weak point. The pixel value changes asymmetrically. When the LSB of cover medium pixel value is equal to the message bit, no change is made. Otherwise, the value is changed from  $2n$  to  $2n + 1$  or  $2n + 1$  to  $2n$ . But the changes from  $2n$  to  $2n - 1$  or  $2n + 1$  to  $2n + 2$  never happen. This asymmetry can be captured by RS steganalysis and Chi square attacks. The methods proposed in this paper are not vulnerable to steganalysis methods like RS steganalysis and Chi square attacks because we are not replacing the LSBs, we are changing the pixel value. As the histograms of original images and stego-images are completely identical; histogram based steganalysis can not detect this steganography. To make all 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.

Capacity means the amount of message that can be hidden. 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. In two, three and four sided side match methods we are hiding data in all the pixels except the first column, first row and last column of the image. As per the sampling arrangement of

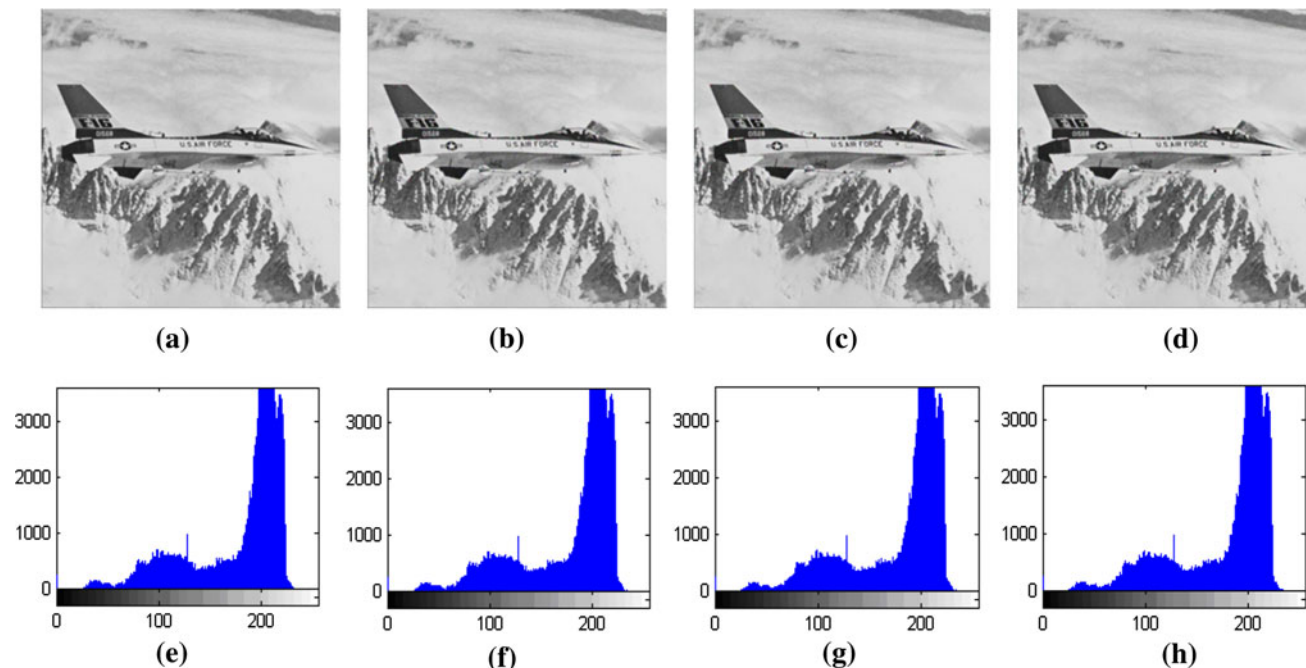
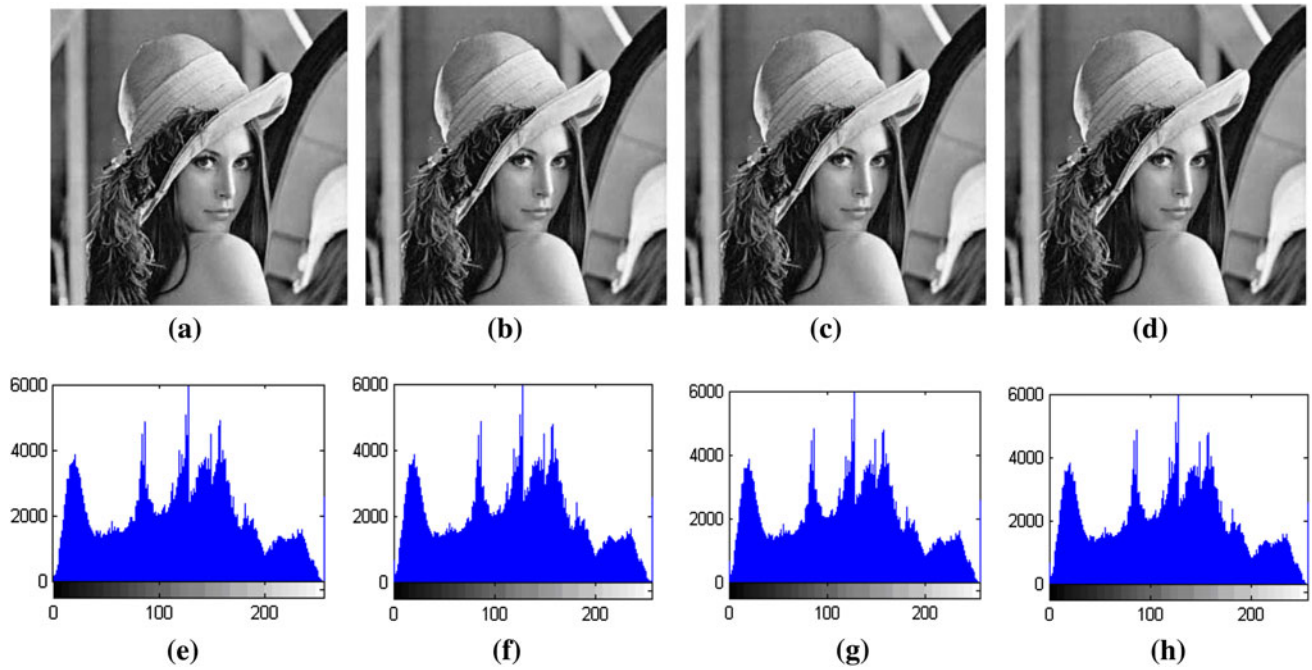
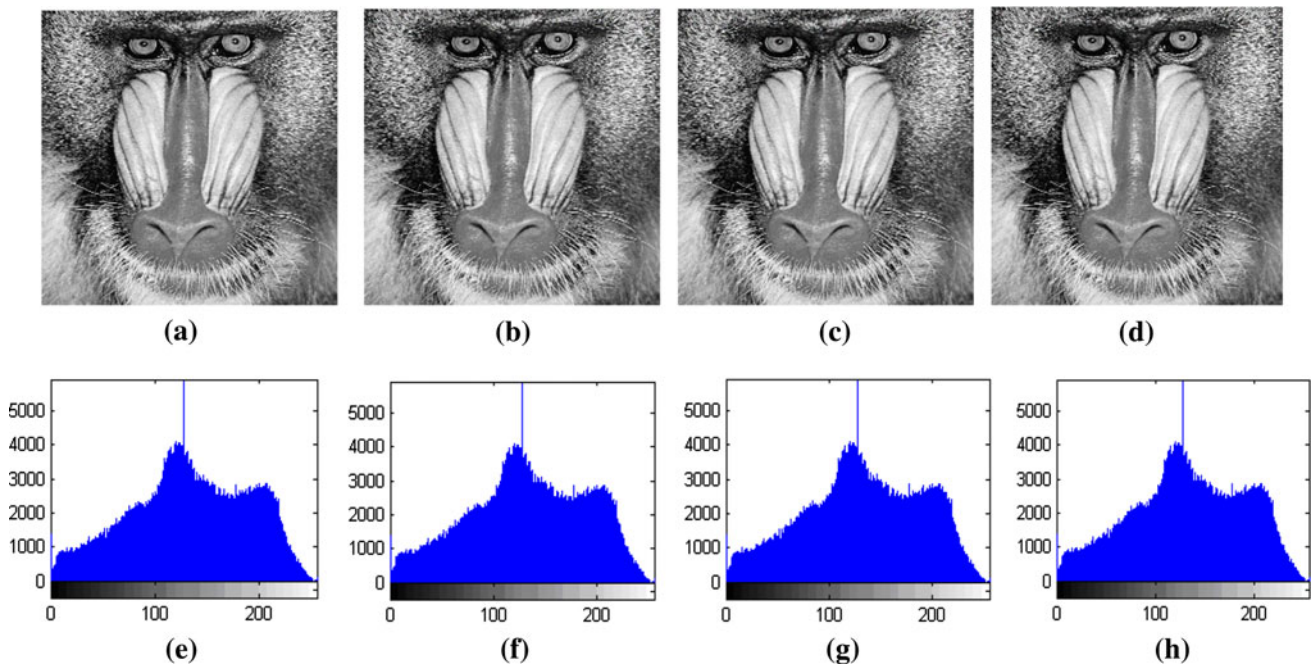


Fig. 4 a Original Airplane image; b, c and d are stego images; e, f, g, h are their histograms



**Fig. 5** a Original Lena image; b, c and d are stego images; e, f, g, h are their histograms



**Fig. 6** a Original Baboon image; b, c and d are stego images; e, f, g, h are their histograms

the pixels in Chang and Tseng's [7] three and four side match methods only 50 % of the pixels can be targeted, but in our methods except the pixels of first row, first column and last column all the remaining pixels can be targeted. Thus our methods can have higher embedding capacity.

Stego-images should not have severe visual artifacts. Without sacrificing the level of security and hiding capacity, if the imperceptibility is higher, then the steganography technique is better. If the resultant stego-image appears innocuous enough, one can say that this requirement is satisfied. It can be observed from Figs. 4, 5 and 6

**Table 1** Observed PSNR values

Image name	Image size (in kb)	Amount of hidden data (in bytes)	Peak signal-to-noise ratio (in dB)		
			Two sided side match	Three sided side match	Four sided side match
Airplane	192	10,240	39.8040	39.7316	40.7624
Lena	535	20,480	44.8240	44.4178	45.3117
Baboon	525	12,288	40.8951	40.9497	42.1573

that the stego-images look very innocuous. We can measure the distortion in the stego-image by peak signal to noise ratio (PSNR) value. The higher is the PSNR value means less is the distortion. If the PSNR value is >40 decibels (dB) it is good. If it falls between 30 to 40 dB it is acceptable. If it falls <30 dB it is not acceptable because the distortion is more. The PSNR values for various images in these three methods are as shown in Table 1. With the same amount of hidden data the distortion is the minimum in four sided side match method as compared to two and three sided side match methods.

#### 4 Conclusion

In this paper we proposed improved versions of two sided, three sided and four sided side match methods by addressing the fall off boundary problem and FIEP. The number of bits embedded in a target pixel is decided depending upon the correlation of the target pixel with its neighboring pixels. The hiding capacities of the proposed methods are comparatively better. After the information is embedded the change in quality of the images are not noticeable. The observed PSNR values are also good. The

distortion is lesser in four sided side match method compared to the two and three sided side match methods. These methods can be used in various occasions where the communication is to be secret.

#### References

1. Cheddad A, Condell J, Curran K, Kevitt PM (2010) Digital image steganography survey and analysis of Current methods. *Signal Process* 90:727–752
2. Jain YK, Ahirwal RR (2010) A novel image steganography method with adaptive number of least significant bits modification based on private stego keys. *Int J Comput Sci Secur* 4(1):40–49
3. Zhang W, Zhang X, Wang S (2007) A double layered plus-minus one data embedding scheme. *IEEE Signal Process Lett* 14(11):848–851
4. Wu DC, Tsai WH (2003) A steganographic method for images by pixel value differencing. *Pattern Recognit Lett* 24(9–10): 1613–1626
5. Zhang X, Wang S (2004) Vulnerability of pixel value differencing steganography to histogram analysis and modification for enhanced security. *Pattern Recognit Lett* 25:331–339
6. Chang KC, Chang CP, Huang PS, Tu TM (2008) A novel image steganography method using tri-way pixel value differencing. *J Multimed* 3(2):37–44
7. Chang CC, Tseng HW (2004) A steganographic method for digital images using side match. *Pattern Recogn Lett* 25(12):1431–1437
8. Kim KJ, Jung K H, Yoo K Y (2008) Image steganographic method with variable embedding length, *International Symposium on Ubiquitous Computing*, p 210–213
9. Zhang H L, Geng G Z, Xiong C Q (2009) Image steganography using pixel-value differencing, *Second International Conference on Electronic Commerce and Security*, p 109–112
10. Li B, He J, Huang J, She YQ (2011) A survey on image steganography and steganalysis. *J Inform Hiding Multimed Signal Process* 2(2):142–172