African Journal of Mathematics and Computer Science Research Vol. 4(9), pp. 286-293, 9 September, 2011
Available online at http://www.academicjournals.org/AJMCSR
ISSN 2006-9731 ©2011 Academic Journals

Full Length Research Paper

New scrambling algorithm for hiding scanned documents

Mohamed Mostafa AbdAllah[1,2] and Gasim Alandjani[1]

[1]Department of Electrical, Communications and Electronics, Faculty of Engineering, Minia University, Egypt.
[2]Department of Electrical and Electronics, Yanbu Industrial College, Royal Commission, Kingdom of Saudi Arabia.

Accepted 18 April, 2011

This paper introduces a new technique to hide scanned document in a digital image. It is a one-to-one matching between the pixels from the cover image to the pixel in scanned documents image. The proposed algorithm is an efficient computer-based steganographic method for embedding secret messages into images without producing noticeable changes is implemented. There is no need of referencing the original cover image while extracting the embedded data from a stego-image. This method utilizes the characteristic of the human vision's sensitivity to gray value variations from smoothness to contrast. Distortions in the cover images are less noticeable because changes in the edge part of the images are less obvious to human observer. The method not only provides a better way for embedding large amounts of data into cover images with imperceptions, but also offers an easy way to accomplish secrecy. This method provides better results as compared to LSB replacement method where the distortions are spread all over the image.

Key words: Steganography, scanned documents, information embedding, secret communication.

INTRODUCTION

Recently, the problem of data hiding and digital watermarking has gained much research interest due to increasing use of digital multimedia (text, image, audio, and video). Data hiding techniques can be used to help identifying and retrieving relevant piece of multimedia data, by embedding information such as keywords on the media. Currently, methods of transmitting secret message through innocuous looking cover mediums called Steganography. Using steganographic techniques we can hide secret information in digital media which has some redundant bits that can be replaced to hide secret data.

Data hiding and digital watermarking deals with an image data type, with a few others considering video and audio data types (Kahn, 1996; Johnson et al., 1998; Anderson et al., 1998). Little work has been reported on data hiding techniques for text or document (Jajodia et al., 1998; Soo-Chang et al., 2003; Wu et al., 2003). To date, two promising data hiding techniques for document image are the line shifting and word shifting methods (Jajodia et al., 1998). Among the two methods, the word shifting scheme offers higher data embedding capacity than the line shifting method. On the other hand, the line shifting method is more resilient to various forms of document processing (such as printing, photocopying, and digitization). For the purpose of document identification and retrieval, high data embedding capacity is preferred. In the spatial domain, image steganography is the simplest technique to embed data in the least significant bit (LSB) of each pixel in the cover image. The fixed-sized method embeds the same number of message bits in each pixel of the cover-image whereas the variable-sized embeds a random number of bits per pixel. Kurak was the first to present such a technique in the early nineties (Kurak and McHugh, 1992). The authors showed how one image can be hidden in another image by replacing the LSB of the cover image by the Most Significant Bit (MSB) of the hidden image. Recently, some steganographic techniques have been reported in Zincheng et al. (2003), Wu et al. (2003), Xinpeng and Shuozhong (2003), Soo-Chang and Jing-Ming, (2003). They showed how data can be directly embedded in the spatial domain of images by directly modifying the absolute values of pixels, or proposed the pixel value differencing (PVD) method by modifying the different values between pairs of adjacent pixels. Using these

*Corresponding author. E-mail: mmustafa@yic.edu.sa.

techniques, more data can be inserted into areas where differences in the adjacent pixel values are large. In the transform domain of an image data can be hidden by modifying the discrete cosine transform (DCT) coefficient values or discrete wavelet transform (DWT) coefficients. These techniques are normally applicable to JPEG images because JPEG images are stored as DCT coefficient values. Another algorithm (F5) proposed by Westfeld (2001) addresses the weaknesses inherent in the Outguess algorithm. This algorithm modified the absolute values of the DCT coefficients instead of modifying its LSB values. It uses matrix encoding and per mutative straddling to reduce the number of steganographic changes. As a result this is resistant to the chi-square test as well as it offer more data embedding capacity compared to Outguess. A more recent work by Sallee presents an information-theoretic method for steganography termed as Model-Based Steganography. It offers high data embedding capacity as well as resistant against statistical attacks (Sallee, 2003; Fridrich, 2004). All the techniques discussed above either try to provide either high data embedding capacity or try to offer resistance against statistical detection but we have found very few researchers who have offered breakthrough thinking on alternate approaches to secure and transmit secret information. In this paper, we use steganography to achieve the goal of disguising the existence of secret communication.

PROPOSED STEGANOGRAPHIC TECHNIQUIE

In this paper we observe that scanned documents contain a lot of redundant data which does not represent any useful information, example,. the white background on which text is written does not offer any useful information. Therefore, even if we do not hide that information it does not make a big difference. Based on this observation we understood that it would be useless to hide this portion in a cover image. We even realized that if we compress the scanned image it would not be as small as an image which only has text without any white background. The point that we want to focus here is the background information is useless and even compressing and hiding it would require a lot of space and doing so would not offer us any useful gain. We only hide that portion of a scanned document which represents textual information and we omit the rest. For doing so, we select an original image (OI) of the same size as the scanned document (SD) and modify those pixels in the original image, which represent textual information in the scanned document. Hence we only hide the portion from the scanned image which represents text. Examples of these images used in the paper are shown in Figure 1. A detailed description of our technique is explained.A generic steganographic technique is described in Figure 2. The sender wants to communicate a “secret message” to a receiver. The message is first “compressed” and then “encrypted”.

The encrypted message can now be secretly hidden in a “cover medium”. A “stego-key” is generated and shared between the sender and the receiver.

This stego-key is used to randomly select and replace the “redundant bits” from the cover media in order to hide the secret message. Redundant bits are defined as those bits in the cover media, which if changed would not change the cover media to a great extent. After embedding is finished the cover media can be transmitted to the receiver. At the receiving end, the “receiver”, having the proper stego-key and decryption key, can “extract” the secret message from cover media. The success of steganography is dependent on the secrecy of the cover media. Once the cover media is public then the success depends on the robustness of the algorithm used.

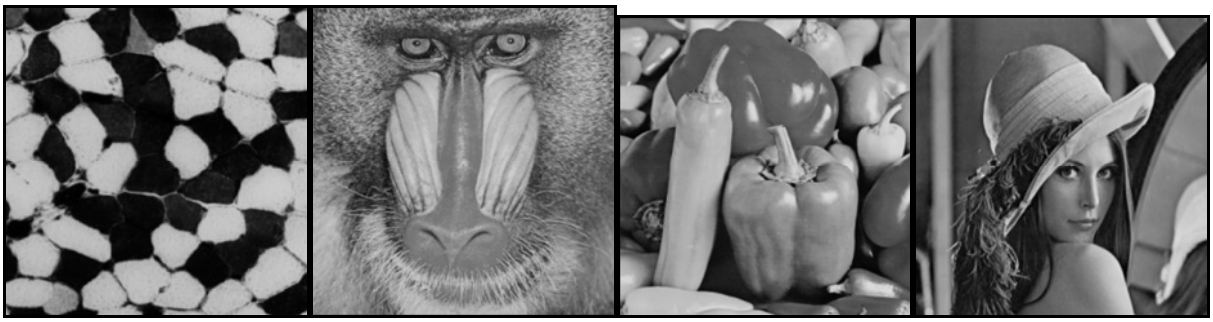
Human visual system

The proposed method for embedding secret messages into a gray-valued cover image uses the fact that, human visual system is having low sensitivity to small changes in digital data. It modifies pixel values of image for data hiding. Cover image is partitioned into non-overlapping blocks of two consecutive pixels. Difference between the two consecutive pixel values is calculated. These difference values are classified into number of ranges. Range intervals are selected according to the characteristics of human vision’s sensitivity to gray value variations from smoothness to contrast. A small difference value indicates that the block is in a smooth area and a large one indicates that it is in an edged area. The pixels in edged areas can tolerate larger changes of pixel values than those in the smooth areas. So, in the proposed method we can embed more data in edged areas than in the smooth areas. The difference value then is replaced by a new value to embed the value of a sub-stream of the secret message. The number of bits which can be embedded in a pixel pair is decided by the width of the range that the difference value belongs to. The method is designed in such a way that the modification is never out of the range interval. This method not only provides a better way for embedding large amounts of data into cover images with imperceptions, but also offers an easy way to accomplish secrecy. This method provides an easy way to produce a more imperceptible result than those yielded by simple least-significant-bit replacement methods. The embedded secret message can be extracted from the resulting stego-image without referencing the original cover image. Experimental results show the feasibility of the proposed method.

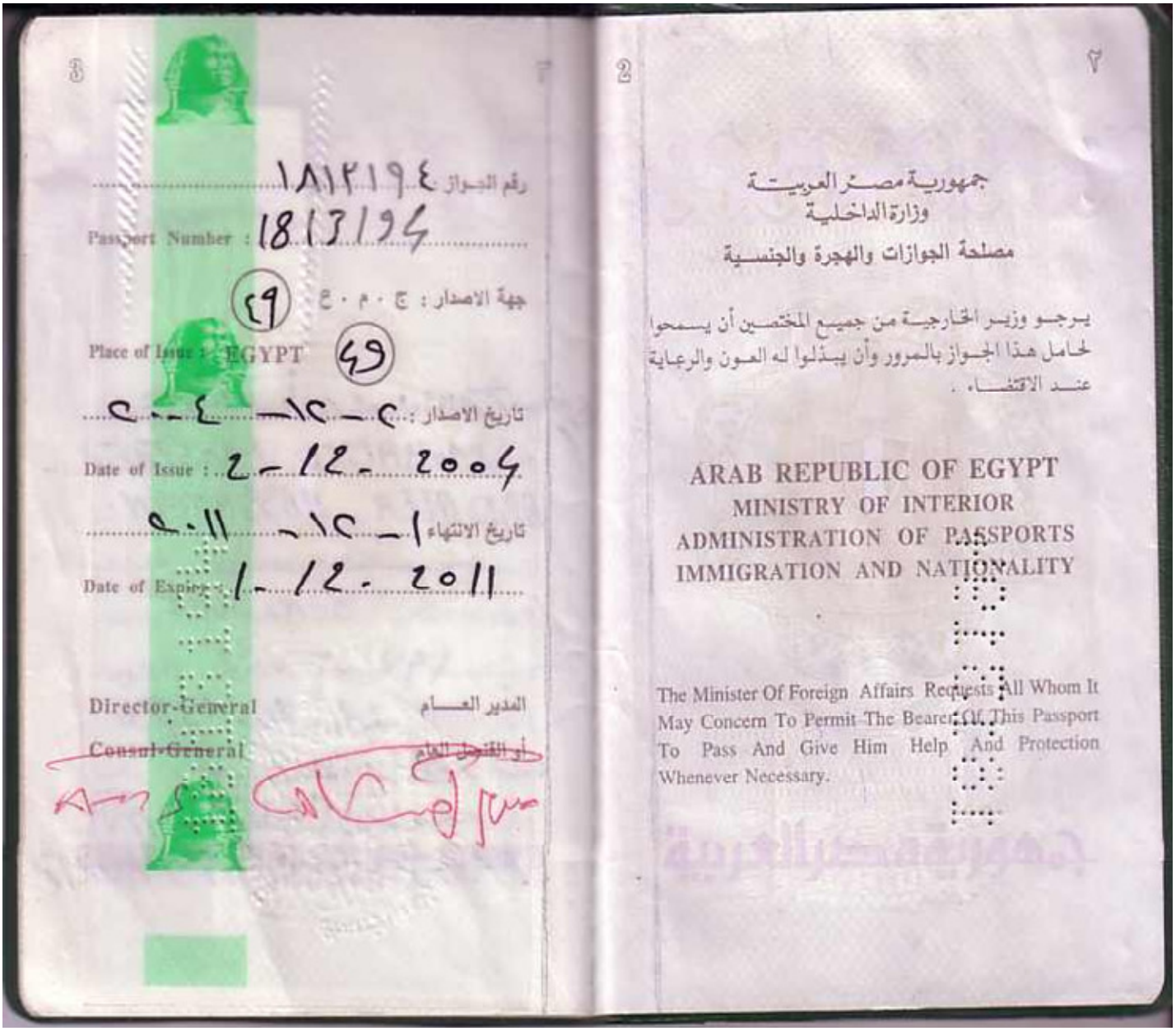
Pixel scrambling technique

Pixel scrambling is a technique to hide scanned document in a digital image. It is a one-to-one matching between the pixels from the cover image to the pixel in SDI. The only limitation on message length is that message length in bits should be smaller than or equal to the number of pixels in the image. However, embedding a large message comparable to the image size increases the likelihood of making detectable changes inconsistent with the dithering algorithm. Detailed analysis of delectability of hidden messages as a function of message length will be part of a future research.

The cover images are 256 gray-valued ones. A difference value d is computed from every non-overlapping block of two consecutive pixels, say P_i and P_{i+1}, of a given cover image. The way of partitioning the cover image into two-pixel blocks runs through all the rows of each image in a zigzag manner, as shown in Figure 3. Assuming that the gray values of P_i and P_{i+1} are g_i and g_{i+1} respectively, then d is computed as $g_{i+1} – g_i$, which may be in the range from -255 to 255. A block with d close to 0 is considered to be an extremely smooth block, whereas a block with d close to -255 or 255 is considered as a sharply edged block. By symmetry, only absolute values of *d* (0 through 255) are considered and classified into a number of contiguous ranges, say R_i where $i = 1,2,\ldots\ldots,n$. These ranges are assigned indices 1 through n. The lower and upper bound values of R_i are denoted by l_i and u_i, respectively, where l_i is 0 and u_n is 255. The width of R_i is $u_i - l_i + 1$. The width of each range is taken to be a power of 2. This restriction of widths



(a)



(b)

Figure 1. (a) Cover image (IO), (b) scanned document (SD).

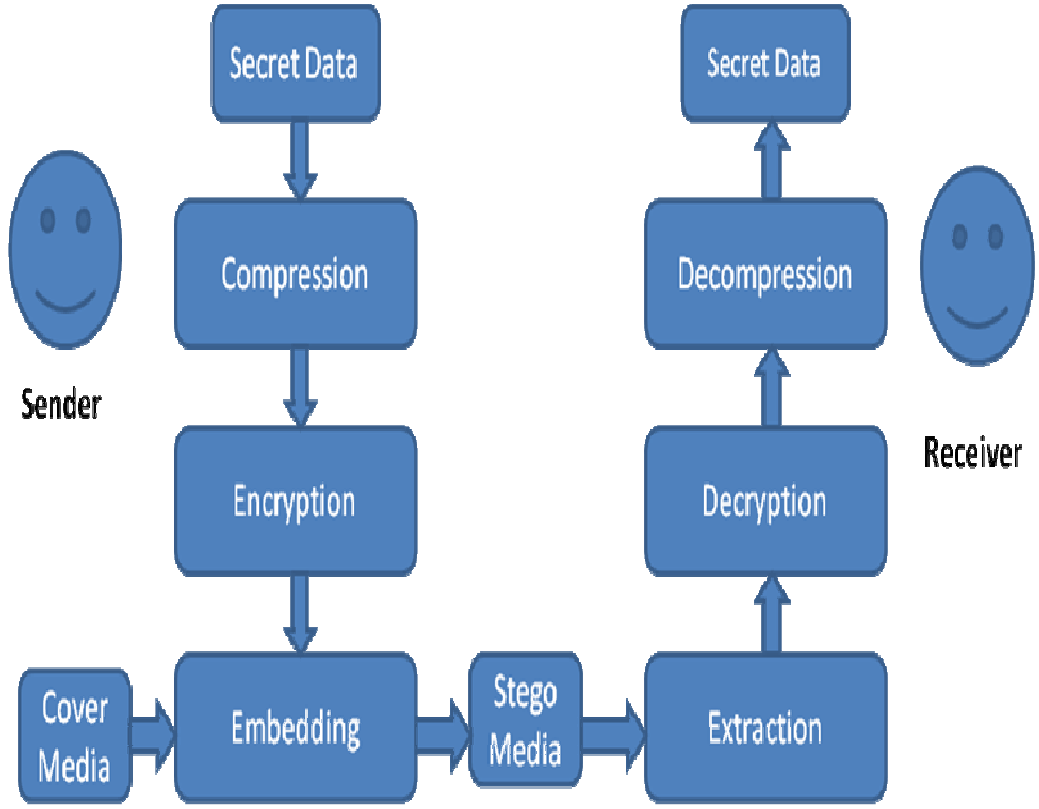


Figure 2. Generic steganographic technique.

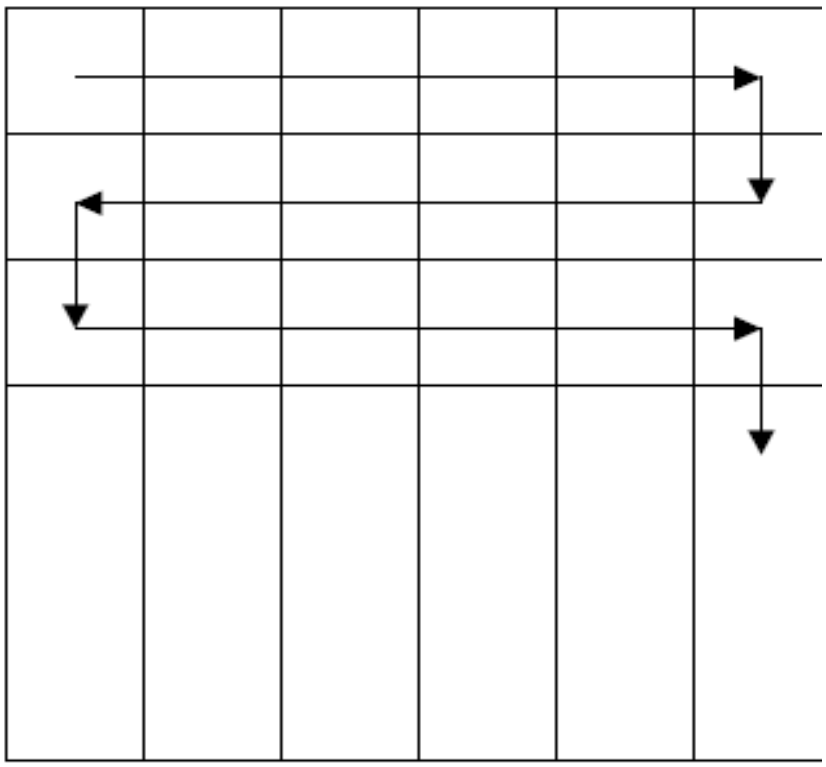


Figure 3. Construction of non-overlapping two pixel blocks by zigzag scanning of the image rows.

facilitates embedding binary data. The widths of the ranges which represent the difference values of smooth blocks are chosen to be smaller while those which represent the difference values of edged blocks are chosen to be larger. That is, ranges are created with smaller widths when *d* is close to 0 and with larger widths when *d* is far away from 0 for the purpose of yielding better imperceptible results. A difference value which falls in a range with index k is said to have index k.

All the values in a certain range (that is, all the values with an identical index) are considered as close enough. That is, if a difference value in a range is replaced by another in the same range, the change presumably cannot be easily noticed by human eyes. Some bits of the secret message are embedded into a two-pixel block by replacing the difference value of the block with one with an identical index, that is, a difference value in one range is changed into any of the difference values in the same range. In other words, the gray values in each two pixel pair are adjusted by two new ones whose difference value causes changes unnoticeable to an observer of the stego-image.

DATA EMBEDDING

Consider the secret message as a long bit stream. Every bit in the bit stream is embedded into the non overlapping two-pixel blocks of the cover image. The number of bits which can be embedded in each block varies and is decided by the width of the range to which the difference value of the two pixels in the block belongs. Given a two-pixel block B with index k and gray value difference d, the number of bits, say n, which can be embedded in this block, is calculated by:

$$n = \log_2 (u_k - l_k + 1) \quad (1)$$

Since the width of each range is selected to be a power of 2, the value of n is an integer. A sub-stream S with n bits is selected next from the secret message for embedding in B. A new difference d' then is computed by:

$$d' = l_k + b \quad \text{for } d \geq 0 \quad (2)$$

$$d' = -(l_k + b) \quad \text{for } d < 0 \quad (3)$$

Where b is the value of the sub-stream S. Because the value b is in the range from 0 to $u_k - l_k$, the value of d' is in the range from l_k to u_k. According to the previous discussions, if d is replaced with d', the resulting changes are presumably unnoticeable to the observer. Then b is embedded by performing an inverse calculation from d' described next to yield the new gray values (g_i', g_{i+1}') for the pixels in the corresponding two-pixel block (P_i', P_{i+1}') of the stego-image. The embedding process is finished when all the bits of the secret message are embedded. The inverse calculation for computing (g_i', g_{i+1}') from the original gray values (g_i, g_{i+1}) of the pixel pair is based on a function f ((g_i , g_{i+1}) ,m) which is defined to be:

$$f((g_i, g_{i+1}), m) = (g_i', g_{i+1}') \quad (4)$$

$(g_i', g_{i+1}') = (g_i - \text{ceiling } m, g_{i+1} + \text{floor } m)$, If d is an odd number;
$(g_i', g_{i+1}') = (g_i - \text{floor } m, g_{i+1} + \text{ceiling } m)$, If d is an even number (5)

where $m = d' - d$, ceiling $m = [m/2]$, and floor $m = [m/2]$ (6)

The above equation satisfies the requirement that the difference between g_i' and g_{i+1}' is d'. It is noted that a distortion reduction policy has been employed in designing Equations (4) and (5) for producing g_i' and g_{i+1}' from g_i and g_{i+1} so that the distortion caused by changing g_i and g_{i+1} is nearly equally distributed over the two pixels in the block. The effect is that the resulting gray value change in the block is less perceptible.

Falling of boundary check

In the inverse calculation, a smaller value of d' produces a smaller range interval between g_i' and g_{i+1}' while a larger d' produces a larger interval. So, (g_i , g_{i+1}) may produce invalid (g_i', g_{i+1}') that is, some of the calculations may cause the resulting g_i', or g_{i+1}' to fall off the boundaries of the range [0, 255]. Although new values may be re-adjusted to the two new values into the valid range of [0, 255] by forcing a falling-off boundary value to be one of the boundary values of 0 and 255, and adjusting the other to a proper value to satisfy the difference d', yet this might produce abnormal spots in contrast with the surrounding region in some cases. To solve this problem, a checking process is used to detect such falling off-boundary cases, and abandon the pixel blocks which yield such cases for data embedding. The gray values of the abandoned blocks are left intact in the stego-image. This strategy helps to distinguish easily blocks with embedded data from abandoned blocks in the process of recovering data from a stego-image, which will be discussed in the next section. It is noted that such abandoned pixel blocks are very few in real applications according to the proposed method.

The proposed falling-off-boundary checking proceeds by producing a pair of ($\hat{g}_i$,$\hat{g}_{i+1}$) from the inverse calculation of the value of the function f ((g_i, g_{i+1}), u_k–d). Since u_k is the maximum value in the range from l_k to u_k, the resulting pair of ($\hat{g}_i$, $\hat{g}_{i+1}$) produced by the use of u_k will yield the maximum difference. That is, this maximum range interval $\hat{g}_{i+1} - \hat{g}_i$ covers all of the ranges yielded by the other ($\hat{g}_i$, $\hat{g}_{i+1}$) pairs. So, the falling-off boundary checking for the block can proceed by only examining the values of ($\hat{g}_i$,$\hat{g}_{i+1}$) which are produced by the case of using u_k. If either of $\hat{g}_i$ or $\hat{g}_{i+1}$ fall off the boundary of 0 or 255, then regard the block to have the possibility of falling-off, and abandon the block for embedding data.

Extracting the embedded message from stego image

The process of extracting the embedded message proceeds by using the same traversing order for visiting the two-pixel blocks as in the embedding process. Each time visit a two-pixel block in the stego-image and apply the same falling-off boundary checking as mentioned previously to the block to find out whether the block was used or not in the embedding process. Assume that the block in the stego-image has the gray values (g_i^* , g_{i+1}^*),and that the difference d^* of the two gray values is with index k. Apply the falling-off-boundary checking process to (g_i^* , g_{i+1}^*) by using f ((g_i^* , g_{i+1}^*), $u_k - d^*$):

$$f((g_i^*, g_{i+1}^*), u_k - d^*) = (\hat{g}_i^*, \hat{g}_{i+1}^*) \quad (7)$$

If either of the gray values of the computed values ($\hat{g}_i^*$, $\hat{g}_{i+1}^*$) falls off the boundaries of the range [0, 255], then it means that the current block was not used for embedding data, or that the block was abandoned in the embedding process. On the contrary, if both of the values ($\hat{g}_i^*$, $\hat{g}_{i+1}^*$) do not fall off the range, it means that some data was embedded in the block. The value *b*, which was embedded in this two-pixel block, is then extracted out using the equation:

$$b = d^* - l_k \quad \text{for } d^* \geq 0 \quad (8)$$

$$= -d^* - l_k \quad \text{for } d^* < 0 \quad (9)$$

Peak noise calculations

When we hide the data in the cover image means we are adding

noise or distortion in that image. It is necessary to calculate Peak signal to noise ratio (PSNR) and root mean square error (RMSE).

$$MSE = \frac{1}{mxn} \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} (\alpha_{ij} - \beta_{ij})^2$$

$$RMSE = \sqrt{MSE}$$

$$PSNR = 10 \log_{10} [(255)^2 / MSE] \quad dB$$

Where: α i,j =Pixel of the cover image in which the coordinate is i,j, β i,j =Pixel of the stego image in which the coordinate is i,j, (m × n) = size of cover and stego image.

EXPERIMENTAL RESULTS

Here four cover images, “Lena”, “Peppers”, “Cell” and “Mandrill” are used, each with size 512 × 512 as shown in Figure 2. Four sets of widths of ranges of gray value differences are used in the experiments. The first experiment is based on selecting the range widths of 2, 2, 4, 4, 4, 8, 8, 16, 16, 32, 32, 64, and 64, which partition the total range of [0 to 255], into [0,1], [2,3], [4,7], [8,11], [12,15], [16,23], [24,31], [32,47], [48,63], [64,95], [96,127], [128,191] and [192,255]. Let us say it as Range 1. The second experiment is based on the use of the range widths of 4, 4, 8, 16, 32, 32, 32, 64 and 64, which partition the total range of [0 to 255], into [0,3], [4, 7], [8,15], [16,31], [32,63], [64,95], [96,127], [128,191] and [192,255]. Let us say it as Range 2. The third experiment is based on the use of the range widths of 8, 8, 16, 32, 64, and 128, which partition the total range of [0 to 255], into [0,7], [8,15], [16,31], [32,63], [64,127] and [128, 255]. Let us say it as Range 3. The fourth experiment is based on the use of the range widths of 16, 16, 32, 64, and 128, which partition the total range of [0 to 255], into [0, 15], [16, 31], [32, 63], [64,127] and [128,255]. Let us say it as Range 4.

The values of the capacities for embedding data by using the cover image and the four sets of range widths are given in Table 1.

Most of the current steganographic techniques, which are discussed in introduction, hide data completely. We argue that we should only hide that amount of data which represents some useful information. As discussed earlier, we designed the SA to achieve high embedding capacity by only hiding that portion of data which represents information. By information we mean the textual portion in the SD. In case of the SDI shown in Figure 4, majority of the image content is purely white background and only a very low percentage is text. As shown in Table 2, we consistently achieve high data embedding capacity because we only hide this fractional part. Since the textual portion in SD corresponds to 5 to 10% of the total image there is no need to store the other redundant information. Such redundant information can be regenerated and this is what the extraction algorithm does. We consistently achieve an embedding capacity of more than 90% compared to the other algorithms. All the current techniques which hide data completely would require at least an embedding capacity more than 25-30 KB for an image of size 512×512. If we consider that every technique uses at least one bit to represent one bit of data (which normally is the case), then it would not be possible for those techniques to hide more than 262,144 bits (512*512 or 31 kB) of information. As shown in the Table 2, we have hidden images within the range of 27 to 58 kB. This has become possible because we only hide a fraction of the information from those images. In the next set of results we identified the threshold value for the range function, that is, we reduced the range of grey level values from 150 to 40 and observed the results. Here we focused on whether the result that we get after reducing the range function, that is, the resultant image, that is, IS(m,n) is legible or not. We changed the range from 0 to 40 till 0 to 150 and observed the results. The results are shown in Table 3. Here we observed that if we change the range from 150 to 40 we can achieve even higher embedding capacity. But at a certain level we have to compromise on quality. There is always a tradeoff between embedding capacity and quality. We gathered some results for the four images shown in Figure 4. We identified how many pixels are required to hide the secret data when the range is fixed at 130, 110,100, 90, 70 and 40. We observed that when we changed the range to 0 to 130 the quality of the output image in all the four cases was still good. But when we reduced it to 90 or 100 some images showed a rather poor quality of output. After identifying the number of pixels modified based on the range we specified, we compared the generated image with the original embedded image and marked it on a scale of 5 based on the legibility factor. The scale we used had five levels to describe the legibility of the image and they were: extremely poor, poor, average, good, excellent. Table 4 shows these results.

EVALUATION AND CONCLUSION

If we look at the stego images distortion are imperceptible to our eyesight. So simply looking at stego image you will not get any idea about secret communication via image.

The enhanced difference images are shown here to indicate the distortions resulting from the data embedding process. Distortions are found on the edges in the images. These distortions are less noticeable because changes in the edge parts of the images are generally less obvious to human eyes. Variable numbers of bits are embedded into the blocks of two pixels. It does not replace the LSBs of pixel values directly; instead, it changes the differences of the two pixel values in a block. We can not find obvious suspicious artifacts on the resulting stego images by simple visual inspection.

Resulting stego images using Ranges 1, 2 and 3 gives imperceptible results. Distortions are observed in stego

Table 1. Hiding capacity using pixel value differencing method.

Cover image	Maximum capacity in bytes			
	Embedding using Range 1	Embedding using Range 2	Embedding using Range 3	Embedding using Range 4
Lena	28883	40497	51692.3	66074
Peppers	27496.9	38775.9	50684.9	65614.8
Cell	23159.6	36683.8	50282.3	59705.9
Mandrill	37189.6	49982	57116	67939

Figure 4. Four images used for testing the SA.

Table 2. Embedding capacity of S.A.

Scanned images (512×512)	A	B	C	D	E	F
	Uncompressed bit format	Compressed JPEG format	No of bits for others	No of bits for P.S.A	Difference bits	% Saving
A	768	44.2	362.056	21.175	340.91	94.15
B	768	53.3	436.633	19.104	417.529	95.62
C	768	27.1	222.003	212.773	212.777	95.84
D	768	58.3	477.593	448.546	448.046	93.81

Table 3. Number of pixels modified by S.A at different range values.

Text images	0-40	0-70	0-90	0-100	0-110	0-130	0-150
A	7876	12834	15156	16170	17179	19212	21175
B	10346	13049	14239	14845	15572	17172	19104
C	15	1389	3407	4384	5370	7252	9226
D	5197	14090	14239	19038	20746	25270	29547

Table 4. Visual judgment of images based on legibility.

Pixel ranges	Image A	Image B	Image C	Image D
0-40	Average	Average	Extra-poor	Extra-poor
0-70	Average	Average	Poor	Poor
0-90	Average	Average	Poor	Poor
0-100	Good	Good	Poor	Average
0-110	Good	Good	Average	Good
0-130	Good	Good	Average	Good
0-150	Excellent	Excellent	Good	Good

images using Range 4, but these distortions are less as compared to conventional LSB replacement techniques. In comparison with LSB replacement method, the pixel value differencing method gives more imperceptible results.

REFERENCES

Anderson RJ, Petitcolas FAP (1998). On the limits of steganography. IEEE J. Selected Areas Commun. (J-SAC) – Special Issue on Copyright & Privacy Protection.

Jajodia S, Johnson N (1998). Steganalysis: The Investigation of Hidden Information." Proc. of the (1998). IEEE Information Technology Conference, Syracuse, New York.

Johnson N, Jajodia S (1998). Steganalysis of Images Created Using Current Steganography Software. Proc. of the 2nd Information Hiding Workshop, Portland, Oregon.

Kahn D (1996). The history of steganography. 1st Information Hiding Workshop. Lect. Notes Comput. Sci., Springer-Verlag, 1174: 1–5.

Pei SC, Guo JM (2003). Hybrid pixel-based data hiding and block-based watermarking for error-diffused halftone images. IEEE Trans. Circuits Syst. Video Technol., 13: 867-884.

Wu DC, Tsai WH (2003). A steganographic method for images by pixel-value differencing." Patt. Recognit. Lett., 24: 1613-1626.

Xinpeng Z Wang S (2003). Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security. Patt. Recognit. Lett., 25: 331-339.