Journal of the Chinese Institute of Engineers

ISSN: 0253-3839 (Print) 2158-7299 (Online) Journal homepage: http://www.tandfonline.com/loi/tcie20

Varying index varying bits substitution algorithm for the implementation of VLSB steganography

Sahib Khan, Nasir Ahmad & Muneeza Wahid

To cite this article: Sahib Khan, Nasir Ahmad & Muneeza Wahid (2015): Varying index varying bits substitution algorithm for the implementation of VLSB steganography, Journal of the Chinese Institute of Engineers, DOI: 10.1080/02533839.2015.1082933

To link to this article: http://dx.doi.org/10.1080/02533839.2015.1082933

Published online: 18 Sep 2015.

Journal of the Chinese Institute of Engineers, 2015
http://dx.doi.org/10.1080/02533839.2015.1082933

Varying index varying bits substitution algorithm for the implementation of VLSB steganography

Sahib Khan[a]*, Nasir Ahmad[a] and Muneeza Wahid[b]

[a]*Department of Computer Systems Engineering, University of Engineering & Technology Peshawar, Peshawar, Pakistan;*
[b]*Department of Electrical Engineering, University of Engineering & Technology Peshawar, Peshawar, Pakistan*

(*Received 6 July 2013; accepted 28 February 2015*)

Variable least significant bits (VLSB) steganography is a pretty powerful and secure technique for data hiding in cover images, having variable data hiding capacity, signal-to-noise ratio, peak signal-to-noise ratio, and mean square error (MSE). This study presents a new algorithm for the implementation of VLSB steganography named varying index varying bits substitution (VIVBS). The VIVBS algorithm is a very secure, high capacity, flexible, and statistically unpredictable mechanism to conceal information in cover images. The method uses a secret stego-key comprising a reference point, and variation of the number of bits to be hidden with varying indices of pixels in the cover image. The secret key adds an extra feature of security to steganography, making it much immune to steganalysis. The VIVBS algorithm is capable of providing variable data hiding capacity and variable key size which can be changed by changing the range of least significant bits used. A data hiding capacity of 43.75% with a negligible MSE 14.67 dB has been achieved using the VIVBS algorithm. For larger data hiding capacity, the MSE and distortion increases significantly which make the existence of information predictable but the key size also increases significantly, making the retrieval of hidden information difficult for the unauthorized person.

Keywords: VLSB steganography; steganalysis; watermarking; discrete cosine transform

1. Introduction

The discipline of data security mainly includes cryptography and steganography. Cryptography transforms ordinary information into nonsense while steganography, furthermore, conceals the existence of information (Tsai et al. 2011). Steganography is an art of hidden writing; the writing that is imperceptible and the presence of data cannot be detected. Steganography is classified on the basis of medium used as cover media, e.g., image steganography and audio steganography (Bhattacharyya, Kim, and Dutta 2012; Dumitrescu, Wu, and Memon 2002; Moon and Kawitkar 2007). A medium having much redundant data is considered to be the most suitable one. An image is a medium with lots of redundant bits which can be used as a cover for data hiding. Lots of work is being carried out in the field of image steganography. Image steganography is classified further into two categories, grayscale image steganography and color image steganography. In this study, special emphasis is given to grayscale image steganography.

Image steganography has been implemented both in the spatial/time domain and in the transform domain, e.g., discrete cosine transform (Song, Wang, and Niu 2012; Walia, Jain, and Navdeep 2010) and wavelet transform (Bhattacharyya and Sanyal 2010). Spatial domain techniques have been implemented to use least significant bits (LSBs), e.g., 4LSBs (Raja et al. 2005; Wang and Wang 2006) and variable least significant bits (VLSB) steganography techniques. The 4LSBs' steganography has an upper bound of 50% information hiding capacity. This is the best one can achieve with this method. To increase hiding capacity of steganography, VLSB steganography was proposed and developed.

2. VLSB steganography

To achieve more than 50% data hiding capacity, a new technique called VLSB steganography was adopted. VLSB steganography is an information concealing technique, utilizing a variable number of bits in a pixel of the cover image to hide data, so varying amounts of information are hidden in different parts of the cover image. This is very different from 4LSBs' steganography (Raja et al. 2005). VLSB steganography seems to be a powerful data hiding technique.

The capacity and signal-to-noise ratio (*SNR*) of steganography are inversely proportional to each other. A trade-off is made to keep both within affordable limits. VLSB steganography also offers a large key size, making it very resistant to steganalysis. The key size and the size of cover image are directly related to each other. To share information using VLSB steganography, only the

*Corresponding author. Email: sahibkhan@uetpeshawar.edu.pk

intended party is provided with the key, to extract the information (Khan, Yousaf, and Akram 2011).

VLSB steganography needs a very professionally well-developed algorithm; it should be capable of hiding more information by keeping *SNR* and mean square error (*MSE*) within an affordable range. It should be strong enough to resist steganalysis by an intruder, so that the information hidden should be imperceptible and the key size of the algorithm should be large to make the recovery of information difficult, in case of detection. Two algorithms/techniques, decreasing distance decreasing bits (DDDB) algorithm (Khan, Yousaf, and Akram 2011), and modular distance technique (MDT) (Khan and Yousaf 2013) were developed to implement VLSB steganography. In decreasing distance algorithms, the number of LSBs used for data hiding varies with the distance of pixels from the central pixel of cover image. As the distance decreases, the number of LSBs used for data hiding decreases. The DDDB algorithm hides more information at borders. In the MDT, the number of LSBs used for data hiding varies with the mode of the distance of a pixel from the reference pixel. In this study, a new technique, named varying index varying bits substitution (VIVBS) algorithm, is presented to implement VLSB steganography.

3. Proposed algorithm

VIVBS is a new algorithm for the implementation of VLSB steganography using variable numbers of bits for data hiding on the basis of the index of the pixel. The number of bits hidden in a pixel, of a specific index, is defined by the key. As each pixel's intensity is represented in an 8-bit binary number, so, there are nine possible values for the number of bits to be hidden. The minimum possible value for the number of bits is 0, which means no hiding, and maximum value is 8 means 100% replacement of cover image pixel with data bits.

VIVBS algorithm processes each pixel of cover image for information hiding. To use the VIVBS algorithm, a pixel's index is calculated using either *x*-intercept or *y*-intercept of pixel position. Then, a table is developed that defines how much data, i.e., bits, will be hidden in a pixel with a specific index/reference number already calculated. A minimum of 0 and maximum of 8 bits can be hidden in a pixel of a specific index number. The maximum value of bits used for data hiding decides the key size, the larger the maximum value of bits, the larger will be the key size and the more secure the hidden information will be. The number of bits assigned to a specific index number is the key. Different index numbers (i.e., Ind1, Ind2, Ind3, Ind4, etc.) may be assigned with different or same number of bits (i.e., N1, N2, N3, N4, etc.) to be hidden. Each pixel is processed using VIVBS algorithm and a number of bits of message are hidden in the pixel according to its index number as given in Table 1.

Now the index for each pixel $P(i,j)$ of cover image in the VIVBS algorithm is calculated. The index value is passed to the table to get the value of number of bits used for hiding information. Then, the value of bits is passed to the VLSB steganography section that hides information in the pixel $P(i,j)$. As the number of bits used for information hiding varies from pixel to pixel on the basis of its index, this is called a VIVBS algorithm. The whole process of VIVBS algorithm is given here in the form of a block diagram in Figure 1.

4. Hiding capacity of VIVBS algorithm

The hiding capacity is the amount of data that can be hidden in a cover file. It is the ratio of total amount of information in bits hidden to the total size of cover in bits. If an image of size "N" is used as a cover and "B_i" bits are hidden in the ith pixel of cover image, then the total capacity "C_t" of covers and total information's bits hidden "B_h" is given as by Equations (1) and (2), respectively.

$$C_t = N \times 8, \tag{1}$$

$$B_h = \sum_{i=1}^{N} B_i. \tag{2}$$

So the information hiding capacity "C" of VIVBS algorithm is given by Equations (3) and (4).

$$C = \frac{B_h}{C_t} \times 100, \tag{3}$$

$$C = \frac{\sum_{i=1}^{N} B_i}{N \times 8} \times 100, \tag{4}$$

where N: size of image; B_h: the number of bits hidden in a pixel; C_t: the total capacity of cover image; C: hiding capacity.

Equation (4) shows that as the number of bits "B_i" hidden in a pixel increases, the hiding capacity increases.

Table 1. Index numbers and assigned number of bits.

Index	No. of bits
Ind1	N1
Ind2	N2
Ind3	N3
Ind4	N4
Ind5	N5
Ind6	N6
Ind7	N7
Ind8	N8
Ind9	N9

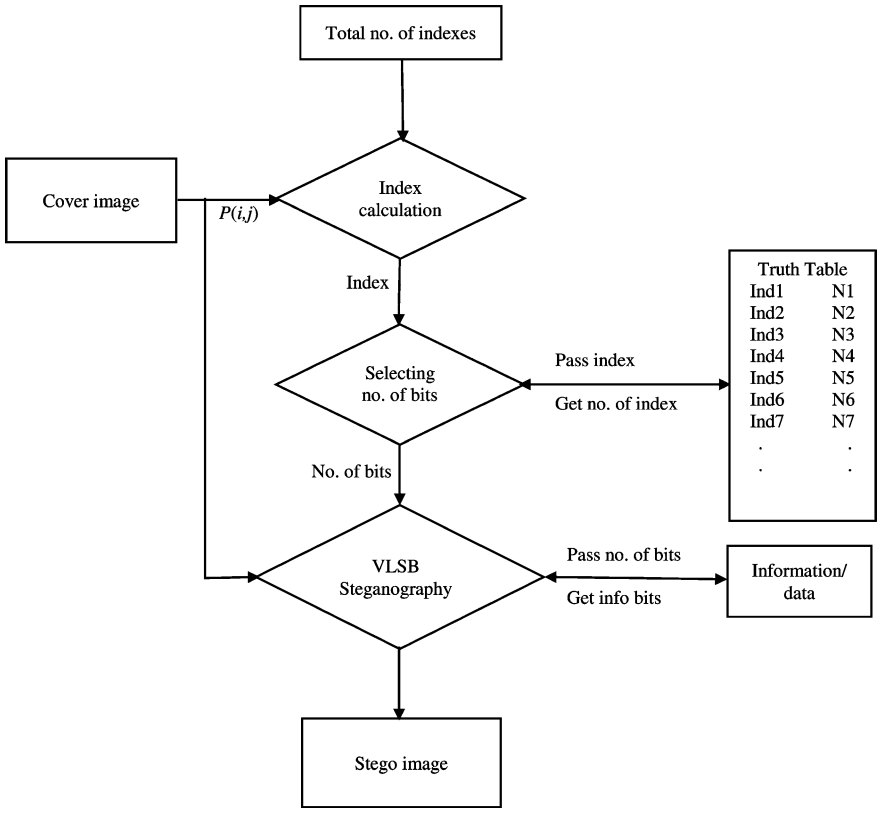

Figure 1. Block diagram of VIVBS algorithm.

5. Key size of VIVBS algorithm

Steganography provides the unpredictability of hidden information in cover image, but in addition, this VIVBS provides an additional feature of security. There are various possible combinations for hiding data in a cover image using the VIVBS algorithm which makes it extremely secure for implementing steganography. Let us say "*n*" numbers of bits are used for information hiding, then the key size "*K*" is given by Equation (5).

$$K = N \times C_1^{(n+1)}. \tag{5}$$

Equation (5) shows that key size is directly proportional to the size of cover image and the depth of the cover image, i.e., the number of bits used, "*n.*" Now if all the 8 bits are used for data hiding using VIVBS algorithm, the maximum key size is achieved. The maximum key size "K_{max}" of VIVBS algorithm is given by Equation (9).

$$K_{max} = N \times C_1^{(8+1)}. \tag{6}$$

As $C_1^{(8+1)} = C_1^9 = 9$ so

$$K_{max} = 9 \times N. \tag{7}$$

As

$$N = r \times c. \tag{8}$$

So

$$K_{max} = 9 \times (r \times c). \tag{9}$$

where K: keys size; K_{max}: maximum key size; r: number of rows of cover image; c: number of columns of cover image.

Equation (9) shows that maximum key size of VIVBS algorithm is 9 times the cover's size. To recover the hidden data, the key plays a vital role and the person, with whom the information is shared, should know the key. Without the key, it is tedious to recover information. The key makes the VLSB steganography quite secure, the greater the key size "*K*", the more difficult it is to reconstruct the hidden information for an intruder.

6. MSE, SNR, and PSNR of VIVBS algorithm

The quality of stego image is analyzed both qualitatively by observation and quantitatively by calculating *MSE*, *SNR*, and peak signal-to-noise ratio (*PSNR*). The hiding capacity and key size for each combination is also calculated to measure the strength of the algorithm. The *MSE*, *SNR*, and *PSNR* are calculated using expressions given below (Gonzalez and Woods 2008; Khan et al. 2013).

$$MSE = \frac{1}{R \times C} \sum_{i=1}^{R} \sum_{j=1}^{C} [Cov(i,j) - Stego(i,j))]^2, \quad (10)$$

$$SNR = 10 \times \log_{10} \left[\frac{\sum_{i=1}^{R} \sum_{j=1}^{C} [Cov(i,j)]^2}{\sum_{i=1}^{R} \sum_{j=1}^{C} [Cov(i,j) - Stego(i,j)]^2} \right], \quad (11)$$

$$PSNR = 10 \times \log_{10} \left[\frac{255^2}{MSE} \right]. \quad (12)$$

7. Implementation results

The implementation process of VLSB steganography with the VIVBS algorithm is explained in Figure 1. The VIVBS algorithm is applied to hide information in a cover image. The cover image and information are shown in Figure 2(a) and (b), respectively. Stego images are obtained for various combinations of numbers of bits "B_i" hidden in the LSBs of cover image pixels. As there are various possible combinations depending on key size, only the results of a few implemented combinations are given here in this study.

A grayscale image has 8 bits per pixel, so we can use the combinations of 2, 3, 4, 5, 6, 7, 8, or 9 LSBs including no hiding condition, which means zero bit substitution. Using two combinations for data hiding results in negligible *MSE* and very high *SNR* and *PSNR*. But the hiding capacity and key size are very small. As the number of combinations is increased, the *MSE* is increased and the *SNR* and *PSNR* are decreased. But along with this, there is a significant increase in hiding capacity and key size.

The resulting stego images of VLSB using the VIVBS algorithm for 2, 3, 4, 5, 6, 7, 8, and 9 combinations of LSBs are shown in Figure 3(a)–(h), respectively. To compare VIVBS algorithm results with the well-known 4*LSBs* steganography, 4LSBs are also applied to the same cover images and the resulting stego images from 4*LSBs* steganography are given in Figure 4. The *MSE*, *SNR*, *PSNR*, hiding capacity, and key size each combination of VIVBS algorithm and 4*LSBs* steganography are given in Table 2.

The results of VIVBS algorithm, listed in Table 2, show that as the combinations of LSBs increase, the capacity for hidden data increases and the *MSE* also increases with the increase in number of LSBs' combinations as shown in the graph given in Figures 5 and 6, respectively. The *SNR* and *PSNR* show a decreasing response with the increase in number of LSBs' combinations and are represented in graphical in Figures 7 and 8, respectively.

The main feature and strength of the VIVBS algorithm is its key size; as given in Table 2, the key size significantly increases with the increase in number of LSBs' combinations. Instead of these combinations, we may use any random combination for data hiding using the VIVBS algorithm which makes it highly secure and largely immune to steganalysis. Although it creates a significant and visible distortion in stego images for 7 and higher combinations of LSBs, the large key size overcomes this and makes it difficult for an intruder to extract the hidden information.

8. Information retrieval

Using a grayscale image as cover, there is a maximum of $8 \times N$ bits capacity available for a user. Eight bits per pixel can be used at maximum for data/information hiding in nine different combinations, i.e., combination of 1, 2, 3, 4, 5, 6, 7, 8, and 9 LSBs. In the previous section, information was concealed in the LSBs of cover images and the resulting stego images of VLSB steganography using VIVBS algorithm for the combinations of 2, 3, 4, 5, 6, 7, 8, and 9 of LSBs were shown in Figure 3. The information hiding process is done on the

Figure 2. (a) Cover image and (b) message image.

Figure 3. (a) Stego image for 2 combinations of LSBs, (b) stego image for 3 combinations of LSBs, (c) stego image for 4 combinations of LSBs, (d) stego image for 5 combinations of LSBs, (e) stego image for 6 combinations of LSBs, (f) stego image for 7 combinations of LSBs, (g) stego image for 8 combinations of LSBs, and (h) stego image for 9 combinations of LSBs.

Table 2. Hiding capacity, *SNR*, *MSE*, *PSNR*, and key size of VIVBS vs. LSBs' combination.

No. of combinations of LSBs	Hiding capacity (%)	*SNR* (dB)	*MSE* (dB)	*PSNR* (dB)	Key size
2	12.50	34.8930	0.0068	69.8319	2 * *N*
3	18.75	28.1845	0.0441	61.6837	3 * *N*
4	25	21.7347	0.2342	54.4342	4 * *N*
5	31.25	16.5082	0.7816	49.2011	5 * *N*
6	37.50	11.4887	3.5493	42.6294	6 * *N*
7	43.75	8.8356	14.6719	36.4659	7 * *N*
8	50.00	7.3876	51.8436	30.9839	8 * *N*
9	56.25	6.5455	132.7809	26.8994	9 * *N*
4LSBs	50.00	12.0921	4.1422	41.9585	1

Figure 4. Stego image of 4LSBs' steganography.

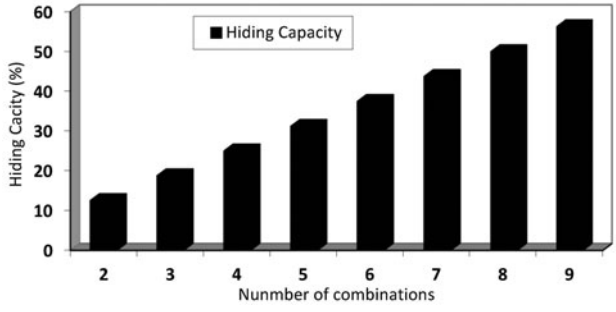

Figure 5. Data hiding capacity vs. LSBs' combinations.

sender side to send secret and valuable information to the intended user, i.e., receiver in a secure manner.

On the receiver side, when a stego image with secret information is received, the process of retrieval of information occurs. This process works in the same manner as that of information hiding process, but in the opposite direction. To retrieve the hidden the information, the stego image is processed pixel by pixel, an index is calculated for each pixel, the number of bits hidden in the LSBs of a pixel having a specific index is found from Table 1 and then information bits are recovered by reading the LSBs. The recovered information bits are arranged to get the message image.

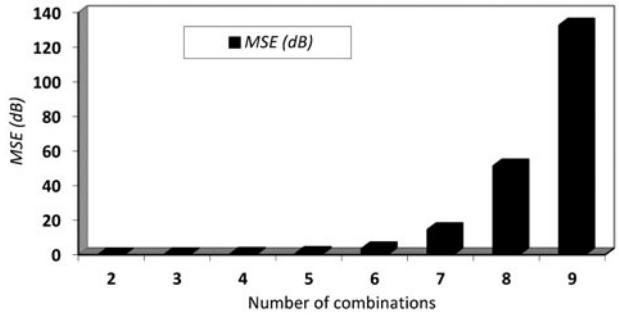

Figure 6. *MSE* vs. LSBs' combinations.

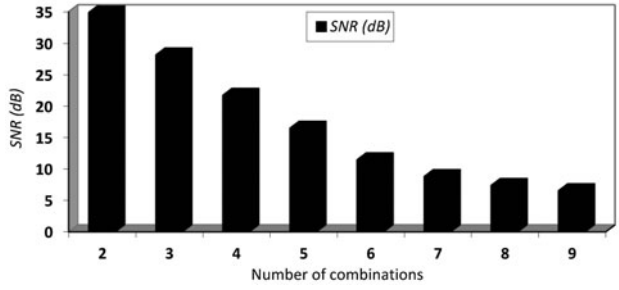

Figure 7. *SNR* vs. LSBs' combinations.

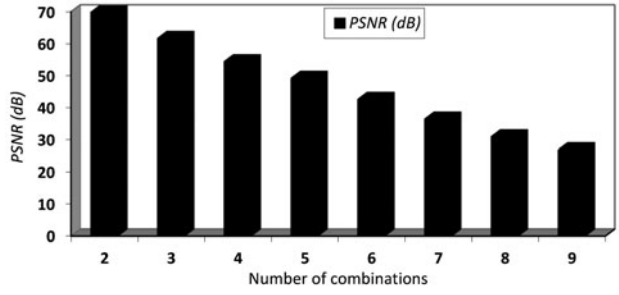

Figure 8. *PSNR* vs. LSBs' combinations.

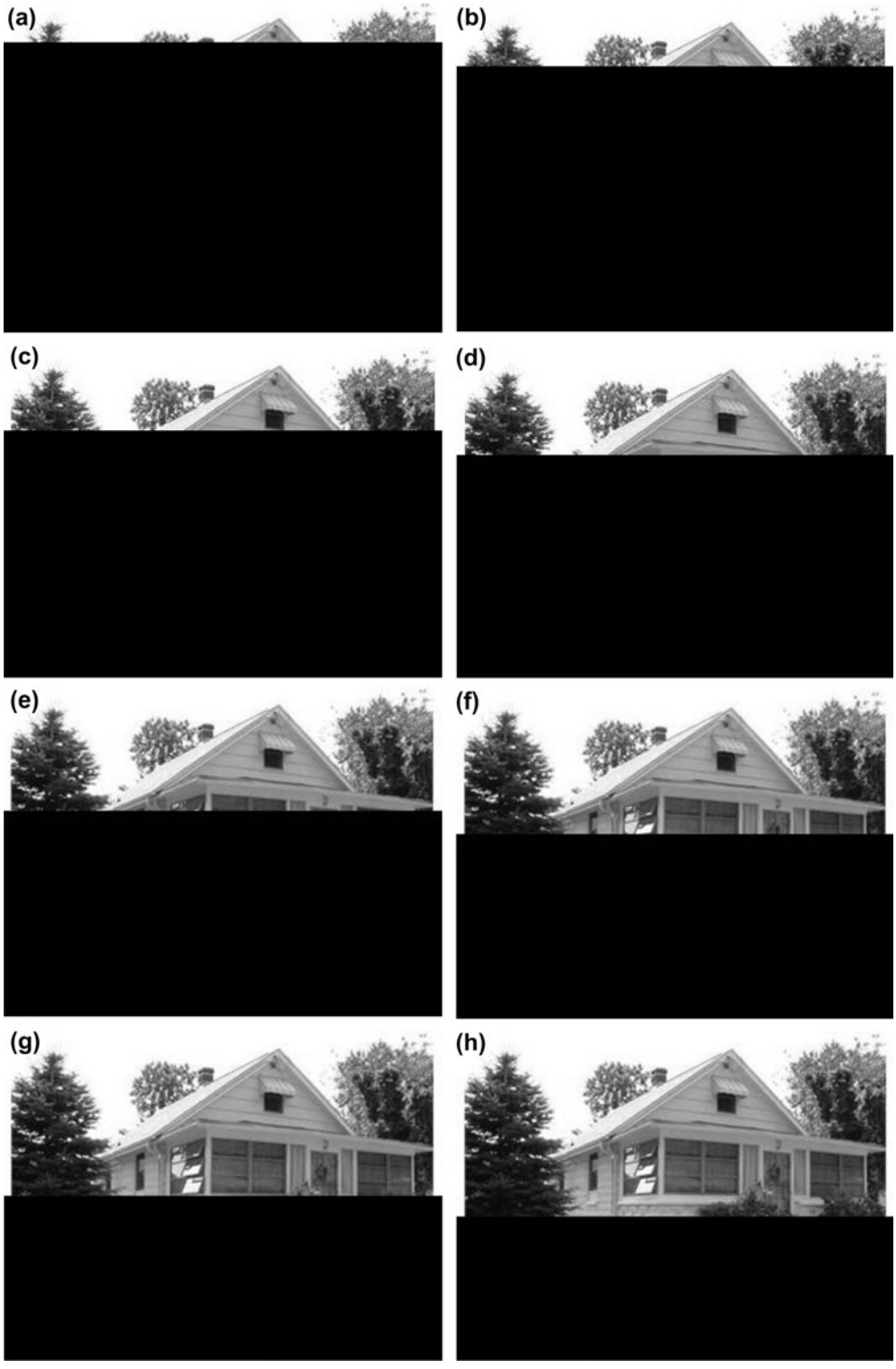

Figure 9. (a) Recovered message (12.50% capacity), (b) recovered message (18.75% capacity), (c) recovered message (25.00% capacity), (d) recovered message (31.25% capacity), (e) recovered message (37.50% capacity), (f) recovered message (43.75% capacity), (g) recovered message (50.00% capacity), and (h) recovered message (56.25% capacity).

Figure 10. Recovered message.

As experimental messages, images were hidden in cover images shown in Figure 2(b) and (a), respectively. Both of these images, considered as message and cover, are of the same size. So, using a combination of 2 bits with hiding capacity 12.5%, only 12.5% of the message image was hidden, and on the receiving side, only 12.5% of the message was recovered. Using other combinations with lager hiding capacities hides more information, and on the receiving side, more information can be recovered. The experimental result showed that all the hidden information is recovered in good health. The recovered messages for 2, 3, 4, 5, 6, 7, 8, and 9 combinations of LSBs are shown here in Figure 9(a)–(g), respectively.

Here, it can be observed clearly that if the image size is larger than the hiding capacity, only a portion of a message can be hidden in a cover depending on the capacity of the LSBs' combination used. In such a case, a message should be hidden in multiple cover images. The same cover image may also be used twice, thrice, etc. to hide the complete message, e.g., the same message image in Figure 2(b) can be hidden using the same cover image in Figure 2(a) twice if the hiding capacity is kept at 50%. And the message recovered is shown here in Figure 10.

9. Comparison

VIVBS algorithm is compared with DDDB algorithm and MDT by hiding the message image of Figure 2(b) in cover image of Figure 2(a). The stego images obtained for VIVBS algorithm, MDT, and DDDB algorithm are shown in Figure 11(a)–(c), respectively, for a fixed hiding capacity of 50.00%. The *SNR*, *PSNR*, *MSE*, time elapsed in hiding "T_h" and time elapsed in recovery "T_r" of VIVBS algorithm are also calculated for each technique as given in Table 3.

The experimental results show that the VIVBS algorithm performed better as compared to DDDB and MDT algorithms in terms of *SNR*, *MSE*, and *PSNR*. The VIVBS algorithm gives a significantly high *SNR* and *PSNR* and a low *MSE*. Although VIVBS takes more

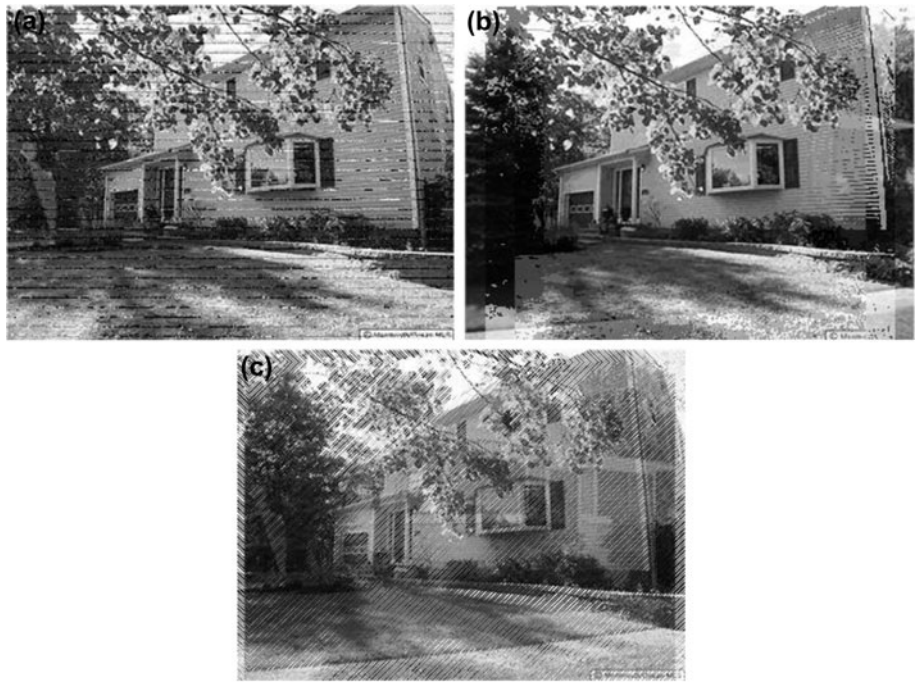

Figure 11. (a) Stego image of VIVBS (50%), (b) stego image of DDDB algorithm (50%), and (c) stego image of MDT (50%).

Table 3. *SNR*, *MSE*, *PSNR*, and elapsed time of VIVBS, DDDB, and MDT.

	Hiding capacity (%)	*SNR* (dB)	*MSE* (dB)	*PSNR* (dB)	Elapsed time "T_h" (s)	Recovery time "T_r" (s)
VIVBS	50.00	7.3876	51.8436	30.9839	133.2998	135.336
DDDB	50.00	−6.0013	111.9193	27.6418	7.4925	8.252
MDT	50.00	−7.5373	122.2027	27.2600	4.9928	5.001

time in data recovery as compared to DDDB and MDT, it is the only approach giving a *PSNR* above 30 dB at 50% data hiding capacity, which is the needed *PSNR* level for most practical applications. The use of VIVBS algorithm is thus the most suitable approach in the cases where high data hiding is required.

10. Conclusion

The increase in the hiding capacity of different steganography approaches is achieved at the cost of quality of the resultant signal. As the hiding capacity increases, the *SNR* and *PSNR* decrease while the *MSE* increases. VIVBS algorithm has a large key size and variable data hiding capability. It performs better than the DDDB and MDT algorithms as it gives higher *SNR*, higher *PSNR*, and lower *MSE* than DDDB algorithm and MDT at a very high level of data hiding. Besides, the key size and time elapsed in retrieving a hidden message using the VIVBS algorithm is significantly larger, making the recovery of data very difficult for an unauthorized person, thus making it suitable for applications where secure exchange of information is needed and there is tolerance of delay, such as military applications.

Nomenclature

MSE	mean square error
SNR	signal-to-noise ratio
PSNR	peak signal-to-noise ratio
N	size of image
B_h	the number of bits hidden in a pixel
C_t	the total capacity of cover image
C	hiding capacity
K	keys size
K_{max}	maximum key size
R	number of rows of cover image
C	number of columns of cover image
T_h	elapsed time
T_r	recovery time

References

Bhattacharyya, D., T. H. Kim, and P. Dutta. 2012. "A Method of Data Hiding in Audio Signal." *Journal of the Chinese Institute of Engineers* 35 (5): 523–528. doi:10.1080/02533839.2012.679054.

Bhattacharyya, S., and G. Sanyal. 2010. "Data Hiding in Images in Discrete Wavelet Domain Using PMM." *Journal of Electrical and Computer Engineering* 5 (6): 359–367.

Dumitrescu, S., X. Wu, and N. Memon. 2002. "On Steganalysis of Random LSB Embedding in Continuous-tone Images." In *Proceeding of International Conference on Image Processing, 2002 Proceedings*, Rochester, 24–28 June 2002, Vol. 3: 641–644. New York: IEEE.

Gonzalez, R. C., and R. E. Woods. 2008. *Digital Image Processing*. 3rd ed. New Jersey, NJ: Prentice Hall.

Khan, S., M. N. Khan, S. Iqbal, S. Y. Shah, and N. Ahmad. 2013. "Implementation of Variable Tone Variable Bits Gray-scale Image Stegnography Using Discrete Cosine Transform." *Journal of Signal and Information Processing* 4 (4): 343–350. doi:10.4236/JSIP.2013.44043.

Khan, S., and M. H. Yousaf. 2013. "Implementation of VLSB Stegnography Using Modular Distance Technique." *Innovations and Advances in Computer, Information, Systems Sciences, and Engineering* 152: 511–525. doi:10.1007/978-1-4614-3535-8_43.

Khan, S., M. H. Yousaf, and M. J. Akram. 2011. "Implementation of Variable Least Significant Bits Stegnography Using Decreasing Distance Decreasing Bits Algorithm." *International Journal of Computer Science Issues* 8 (6): 292–296.

Moon, S. K., and R. S. Kawitkar. 2007. "Data Security Using Data Hiding." In *International Conference on Computational Intelligence and Multimedia Applications*, Sivakasi, Tamilnadu, 13–15 December 2007, Vol. 4: 247–251. New York: IEEE.

Raja, K. B., C. R. Chowdary, K. R. Venugopal, and L. M. Patnaik. 2005. "A Secure Image Steganography Using LSB, DCT and Compression Techniques on Raw Images." In *Third International Conference on Intelligent Sensing and Information Processing (ICISIP)*, Bangalore, 14–17 December 2005: 170–176. New York: IEEE.

Song, X., S. Wang, and X. Niu. 2012. "An Integer DCT and Affine Transformation Based Image Steganography Method." In *Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, Piraeus, 18–20 July 2012: 102–105. New York: IEEE.

Tsai, C. T., C. Liaw, Y. H. Liao, and C. H. Ko. 2011. "Concealing Information in Image Mosaics Based on Tile Image Features." *Journal of the Chinese Institute of Engineers* 34 (3): 429–440. doi:10.1080/02533839.2011.565618.

Walia, D. E., P. Jain, and N. Navdeep. 2010. "An Analysis of LSB & DCT Based Steganography." *Global Journal of Computer Science and Technology* 10 (1): 4–8.

Wang, C. M., and P. C. Wang. 2006. "Data Hiding on Point-sampled Geometry." *Journal of the Chinese Institute of Engineers* 29 (3): 539–542. doi:10.1080/02533839.2006.9671149.