An Optimum Modified Bit Plane Splicing LSB Algorithm for Secret Data Hiding

M. Naseem
Department of Computer Engineering,
Sir Syed University of Engineering and Technology (SSUET), Karachi - Pakistan

Ibrahim M. Hussain
Department of Computer Engineering,
Sir Syed University of Engineering and Technology (SSUET), Karachi - Pakistan

M. Kamran Khan
Department of Computer Engineering,
Sir Syed University of Engineering and Technology (SSUET), Karachi - Pakistan

Aisha Ajmal
Department of Computer Engineering,
Sir Syed University of Engineering and Technology (SSUET), Karachi - Pakistan

ABSTRACT
In this paper, we propose an algorithm called Optimum Intensity Based Distributed Hiding (OIBDH) for secret data hiding inside cover images. The algorithm is a modified version of Bit Plane Splicing LSB technique with better hidden capacity and improved embedding process. The proposed algorithm outperforms Bit Plane Splicing LSB technique as more data can be hidden without degrading the quality of the cover image. Furthermore, both algorithms are tested using entropy curves and results show that OIBDH has lower absolute entropy difference compared to Bit Plane Splicing LSB technique in all the tested images.

General Terms
Algorithms, Security, Image processing.

Keywords
Cover image, Entropy, Hiding capacity, LSB, Steganography.

1. INTRODUCTION
Information hiding and watermarking has been a major research topic for the last few years due to its massive application in various fields. Such applications include but not limited to user authentication for security purposes, digital media protection, copyright reservation and secret information transmission over secured channel. Furthermore, the advent of the *Internet* has resulted in many new challenges regarding information transmission which includes electronic advertising, real-time video and audio streaming and web publishing [1]. The onset of the internet has brought *steganography*, a technology combining information hiding and watermarking, into a real consideration to encounter challenging problems associated with the internet and digital media transmission.

Steganography is all about concealing and hiding the existence of the information to be hidden and providing a secret communication between sender and receiver [2]. Steganography can be split into two types, fragile and robust. In fragile steganography [3], information is embedded into a carrier called 'Cover' without being detected or retrieved by an attacker or unauthorized user as shown in Fig. 1. If the fragile data is modified or destroyed, then the secret information contained within the cover will be destroyed as well without being detected. On the other hand, robust steganography or sometimes known as digital watermarking aims to embed information into a cover while providing detection and protection capabilities if an attempt is made to destroy the transmitted data [4].

Throughout this paper we use the following terms to describe various objects and entities being involved in our work:

Secret data: refers to the data or information to be hidden inside the cover object. Since we are dealing with digital data, hence the data is in the form of bits.

Secret data size: The amount of secret data to be hidden.

Cover object: refers to the object or file in which the data is embedded. The cover object could by a file, multimedia object (video or voice) or simply an image. Throughout this paper, the cover object we use is image.

Stego object: refers to the embedded cover object. In case of an image we refer to it as stego image. Image steganography is highly multidisciplinary field combining image and signal processing with cryptography, communication theory, coding theory, signal compression, and the theory of visual perception. It is a major research area in this field and has received a tremendous interest from industry and military.

Stego algorithm: refers to the algorithm, technique and process by which the secret data is embedded into the cover object.

In robust steganography or watermarking, usually the cover object and secret data are related to each other. For instance, a digital movie (cover) might simply identify the name or version (secret data) of the movie. Furthermore, watermarks often are not perceptible by humans at all, but rather are designed to be detected and recovered only by programmable machines. Such watermarks are called *invisible digital watermarks* [5,6]. On the other hand, in perceptible watermarking the watermark is intentionally perceptible or visible to a human observer [7]. This is usually used for copyright and authentication marking (e.g. logo of a news channel during exclusive video and breaking news transmission).

In case of fragile steganography, data is hidden inside the cover without being noticed by an attacker. This technique has many applications especially in secured data transmission

environment. This includes information related to military and sensitive organizations.

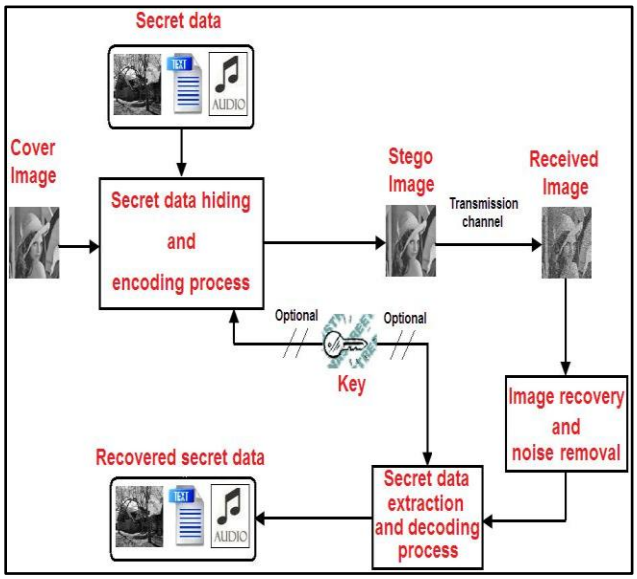

Fig 1: A block diagram of information hiding and steganography process

Many fragile algorithms and techniques have been introduced and proposed for secret data transmission. Most fragile algorithms are simple but less robust than robust steganography algorithms. Fragile algorithms can be divided into two categories, adaptive and non-adaptive. In non-adaptive, in case of image steganography, modifications due to secret data embedding are uncorrelated with image features. On the other hand, in adaptive algorithms modifications are correlated with the image content (features). Most transform based algorithms which are commonly used for information hiding are adaptive in nature.

As an example of such algorithms and techniques, in [8], data hiding is achieved using integer wavelet transform without distorting the cover image. This algorithm hides data into one (or more) middle bit-plane(s) of the integer wavelet transform coefficients in the middle and high frequency sub-bands. Discrete Cosine Transform (DCT) is another promising technique used for both data hiding and image compression [9]. Channel coding and modulation schemes such as turbo codes [10] and convolutional codes using soft-decision decoding are also used for watermarking and steganography. All these mentioned algorithms do not directly modify the contents of an image or pixel values. On the other hand, some algorithms use direct methods to modify the contents of an image feature or pixel. Bit Plane Splicing LSB technique is an example of such algorithm used to modify the pixel value (gray level or color intensity) of an image pixel by pixel [11-14]. This technique is very efficient and simple method for data hiding inside the lower bit planes of an image. Low invisibility is the major drawback of this technique. As more secret data is inserted into the bit planes the cover image becomes more degraded and artifact patches appears in different areas of the image [11]. Hence the perceptual transparency which is a requirement of a strong fragile steganography is poor when Bit Plane Splicing LSB technique is applied on a cover image. In addition, this technique is non-adaptive and treats each and every pixel in an image equally when it comes to secret data hiding. In this paper we have proposed a modified version of the Bit Plane Splicing LSB technique which is not only adaptive in nature but also has more data capacity and better invisibility performance than the normal Bit Plane Splicing LSB technique.

The improvement presented in our work is not merely based upon the human visual system but rather based upon a degradation information theory based parameter which we have used throughout our experiments called the *Entropy*.

Rest of the paper is organized as follows: in section 2 an overview of the normal Bit Plane Splicing LSB technique and its performance is presented. Our proposed algorithm is given in section 3. Test results of the proposed algorithm along with different issues associated with it is discussed in section 4. Finally conclusion is made in section 5.

2. BIT PLANE SPLICING LSB TECHNIQUE

A gray scale digital image consisting of $m \times n$ pixels such that each pixel is represented by r bits has a size of $m \times n \times r$ bits. A common value of r is 8. In case of $r = 8$, a pixel can have an integer value between 0 (black pixel) and 255 (white pixel) which represents the gray level intensity of a pixel. Furthermore, such an image can be split into 8 layers such that each layer is considered to be a separate image. For instance if the pixel intensity in an image is 200 for which the binary representation in 8 bits is 11001000, then the 8 planes of this pixel starting from the least significant bit or plane are 0, 0, 0, 1, 0, 0, 1, 1. For $m \times n$ size image, each plane is going to be of size $m \times n$ and each plane can be considered to be an image having pixel values of either 0 or 1 (i.e. each plane is a binary image). This concept is illustrated in Fig. 2. In this figure each plane is displayed as a separate image to find out the contribution of each plane in building the overall image. It is evident that most of the data is contained in plane 8 and has a major contribution in building the original image. As we go towards the LSB plane, the information contained decreases. Infact the lower half plane has least contribution. We can also see that as we eliminate each plane starting from the LSB plane, degradation occurs more and quality of the image becomes poor. Hence eliminating the lower half planes has less impact in quality degradation of the actual image than eliminating the upper half planes.

Based upon these facts the Bit Plane Splicing LSB technique for information hiding was introduced that hides information or secret bits in the lower bits by replacing the lower bits of pixels in the cover image with secret bits [12]. The replacement of bits takes place pixel wise in a sequential manner either horizontally or vertically. When a particular plane is fully replaced by secret bits the next immediate higher plane is accessed as elaborated in Fig. 3. Since the replacement takes place pixel wise this method does not differentiate between pixels' attributes, sensitivity, location and intensity. It treats a pixel as a bundle of bits and thus the method is static in nature.

To further see what happens when each plane of the cover image is replaced by random bits, we perform a test on camera man image as shown in Fig. 4. In this test we replace each plane totally by random bits of a size $m \times n$ bits. After each replacement, the image is displayed to see the impact on the

cover image. It is clear that till plane two, there is almost imperceptible degradation in the cover image. But beyond second plane, degradation in the form of artifacts can be seen especially in the fourth and fifth plane. This shows that Bit Plane Splicing LSB technique has limited data insertion capabilities and degrades the cover image severely. Generally, regardless of size and type of image Bit Plane Splicing LSB technique has hiding capacity not beyond the second plane. This puts a huge limitation over the implementation of this technique especially when the secret data is greater than two plane size of the cover image.

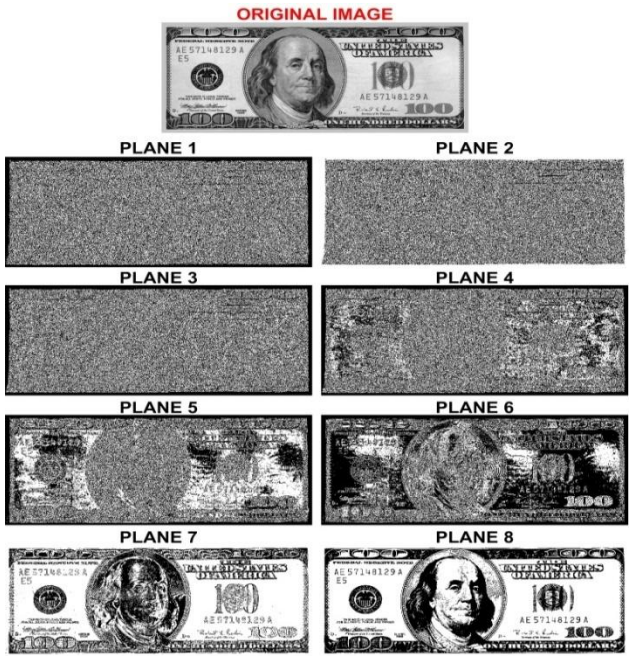

Fig 2: Contribution of the eight bit planes in constructing the original image

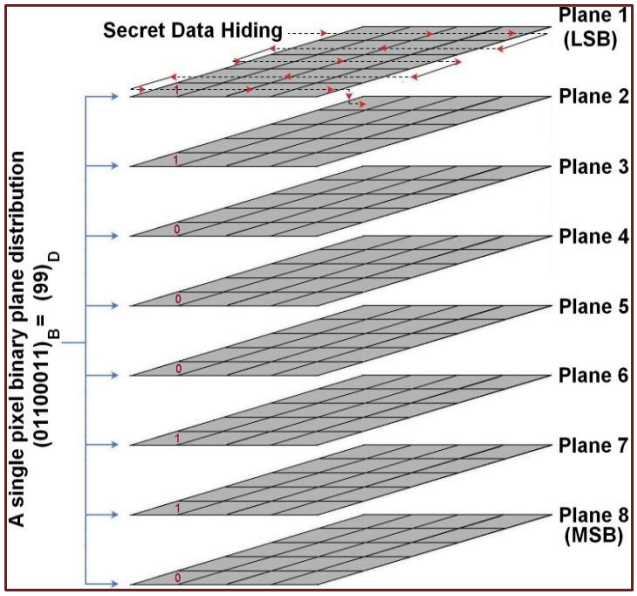

Fig 3: Plane distribution and data insertion method in Bit Plane Splicing LSB technique

To further evaluate the performance of Bit Plane Splicing LSB technique, we need to check its hiding capacity and the amount of degradation in the quality of cover image after data hiding process. As a general rule, quality degradation of the cover image is directly proportional to the size of secret data to be hidden. For this purpose we use *Entropy* and *Entropy* versus *Secret Information Size* curves for performance evaluation of this algorithm and our proposed algorithm [15].

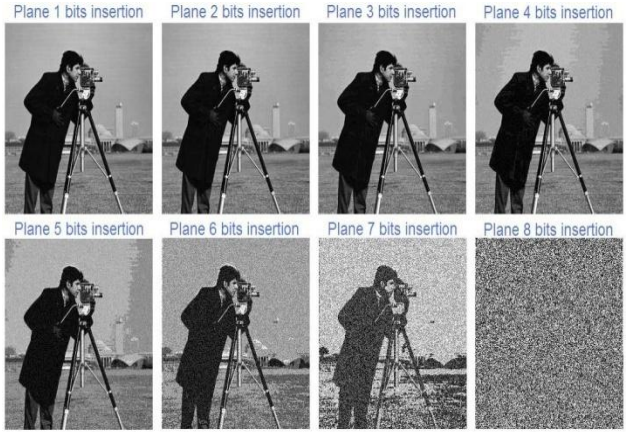

Fig 4: Image degradation due to Bit Plane Splicing LSB technique plane wise staring from plane 1 till plane 8

It is well known that the entropy of an image $f(x,y)$ is given as [16]:

$$Entropy = \sum_{i=0}^{l-1} P_i \log_2 P_i \qquad (1)$$

where P_i indicates the histogram or probability value of a particular pixel intensity in $f(x,y)$ and $l = 2^r$ is the total intensity values. The size of an image has no effect on the entropy whereas the probability of occurrence of the intensity levels of pixels has a major role in producing different entropy values. This concept is shown in Fig. 5 in Lena's image. The image is of size 300×300 pixels and the intensity range is from 0 to 255. Fig. 5 shows that at level 0 (means when no modification is done on the image) the entropy value using (1) is around 7.2738 which is the maximum value. Now as we darken or brighten by a level (e.g. level 1 means all pixel intensities are increased by 1), entropy value decreases. Note that we have increased or decreased the intensity of each pixel by the same amount and truncated those intensity values out of bound (i.e. 0 and 255). It is evident that as we change the intensity of an image in this manner, the entropy value decreases till it becomes 0 (i.e. when all pixels are either black or white and the image conveys no information according to information theory). Similarly if we suppose that the secret data to be hidden is random in nature and assumed to be an external noise with a specific probably density function then as the size of the secret data increases, the absolute entropy difference between the original image entropy and the entropy of the image after secret data insertion also increases. In Fig. 6, the same camera man image is used to support this fact. In this case we have chosen the 'salt-pepper' noise as secret data which is inserted in the cover image. Fig. 6 (a) shows a low level noise and Fig. 6 (b) is a high level noise both inserted into the cover image for which the entropy differences are 0.0311 and 0.5116 respectively. This shows that more we insert data

into the cover image, more changes occur in the entropy value of the cover image.

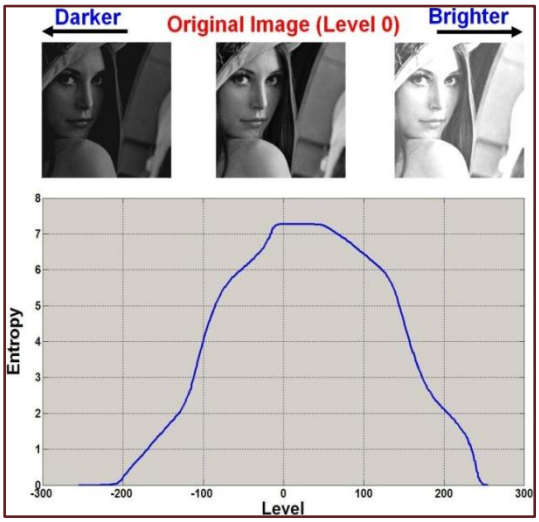

Fig 5: The effect of uniform intensity change in an image on the entropy value

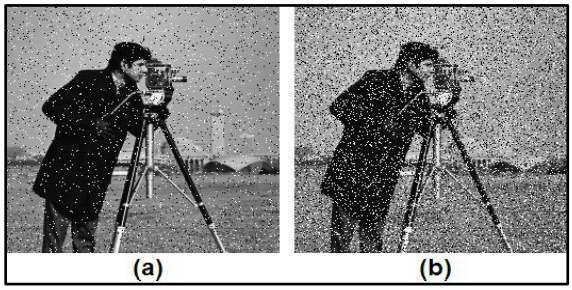

Fig 6: The effect of random noise insertion in an image on the entropy value

For Bit Plane Splicing LSB algorithm, entropy curve shown in Fig. 7 (f) is generated in the same manner by calculating the original entropy of the cover image then after each data insertion (in bits) as indicated earlier pixel wise and plane after plane in a sequential manner the entropy of the stego image is calculated after which the absolute entropy difference is calculated. The experiment is done on camera man image shown in Fig. 7 (a). Initially when there is no secret information the stego image is same as the original cover image and the absolute entropy difference is zero. Now as the size of secret data increases the absolute entropy difference also increases. Or in other words, the quality of the image degrades as the size of secret data to be hidden increases. Since the image size in bits is $256 \times 256 \times 8$ with 8 planes, the secret data size is from 0 bits (i.e. no secret data) to 524288 bits (i.e. when secret data size is same as cover image size). Now to see the effect in the quality of image as the size of secret data varies, from Fig. 7 (f) we have selected 8 points from the curve corresponding to 8 different sizes of secret data indicated by red circles. These points correspond to the 8 planes of the cover image. For instance, the first secret data size is $256 \times 256 \times 1 = 65536$ bits which means that the LSB plane or the lowest plane of the cover image is replaced by the secret data. Table I shows 8 entries of secret data sizes, the corresponding plane being replaced and the absolute entropy difference whereas Fig. 7 (b-e) shows four of the stego images resulted in case of planes 2, 3, 4 and 7 respectively. Visually we can see that till plane 2 the hiding secret data is tolerable as the quality of the cover image does not degrade but as we move from plane 2 to plane 3, visual degradation especially in uniform intensity areas (e.g. sky) is evident as shown in Fig. 7 (c). Moving towards plane 4 in Fig. 7 (d) i.e. when 50% of the data is hidden in the lower half of the cover image, extreme degradation occurs and the artifact effect is prominent even in dark areas (e.g. coat of camera man). Beyond plane 4 the image becomes unrecognizable. This experiment shows that generally speaking for Plane Bit Splicing technique, the maximum capacity of secret data to be hidden in a cover image is approximately between 25 to 35% of the cover image size. This range could slightly vary from image to image with different shades and intensity levels of pixels.

Beside the above mentioned degradation, another visual degradation occurs when the secret bits expand partial areas of the plane due to its sequential insertion method. For instance assuming that the insertion takes place starting from the top left pixel and continues column wise, now if the cover image size is (600 × 600 × 8) bits and the secret data size is (600 × 600 × 4.5) then the visual degradation after hiding the secret data occurs more severely on the left half section of the stego image because the secret data expands the left half of the fifth plane where as the right half section of the fifth plane remains intact without modification. The overall impact results in a stego image with more visual degradation on the left side compared to the right side as it is evident in Fig. 8. The left shades in Fig. 8 (b) (black, white and gray) are more degraded than their counterparts on the right portion since secret data expands only half of the third plane.

In summary the major disadvantage of Bit Plane Splicing LSB technique is that it degrades the cover image if the size of secret data is large. This results in an attacker monitoring the transmission to suspect a hidden object or data inside the cover image which might further trigger an unwanted and harmful action by the attacker. Our proposed algorithm handles this problem by reducing the degradation caused by hiding data which is discussed in the next section.

3. OPTIMUM INTENSITY BASED DISTRIBUTED HIDING (OIBDH) ALGORITHM

In this section we propose our algorithm called Optimum Intensity Based Distributed Hiding (OIBDH). In this algorithm as in case of Bit Plane Splicing LSB the lower bits are used to hide secret bits but now in a different manner. Instead of hiding bits pixel by pixel and plane after plane the bits are hidden based on the color intensity of a pixel. We have chosen two ranges of intensity levels. Assuming that l = 256 (i.e. 8 planes), for the intensity of pixels between 0−32, the lowest 3 bits are used for hiding secret bits where as for pixels having intensity of more than 32, only the lower 4 bits are used. By performing heuristic testing with different range of values and searching algorithm we have come up with this optimum range. This range is a near optimum range and causes least degradation when it comes to over all image. This distribution is done based on the weight of

pixel and its degradation severity (i.e. the human visual perception due to transition of pixel intensity from one level to another). In addition, since bright intensity levels are more sensitive when transition takes place and dark levels are less sensitive, we have chosen less number of bits for hiding in brighter pixels and higher in darker pixels. Furthermore the method of hiding bits is different in case of OIBDH. In case of Bit Plane Splicing LSB a single secret bit and its very adjacent bit are hidden in adjacent pixels since the secret data are embedded sequential wise. But in case of OIBDH, if the first bit is hidden in plane *q* and in pixel *w* then the next bit is hidden in the same plane but in any pixel in the cover image and not necessarily in the adjacent pixel as the next pixel with same intensity value could be located anywhere in the image. In this way the secret data is distributed throughout the image based upon the color intensity of pixels which is random for every image. In addition, those pixels having intensity levels between 0−32 are used first for data insertion followed by the second range of pixels. Note that not only the bits are hidden randomly in pixels but also the planes are traversed in a non sequential manner and not plane by plane as in the case of Bit Plane Splicing LSB technique. To elaborate the steps through which the algorithm works, a flow diagram of the proposed algorithm is given in Fig. 9 at the sender's end.

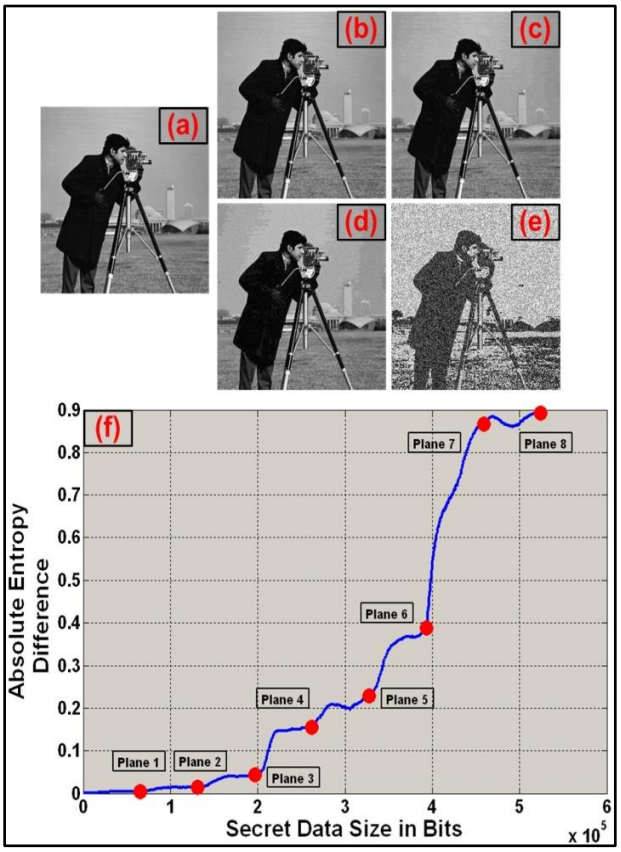

Fig 7: (a) original cover image (b-e) image degradation due to data hiding using Bit Plane Splicing LSB technique when data size equals to plane 2, 3, 4 and 7 respectively. (f) Entropy curve of camera man image when Bit Plane Splicing LSB technique is applied

Table 1. Secret data sizes and the corresponding absolute entropy difference values taken from the entropy curve shown in Fig. 7 (f)

Secret Data Size (bits)	The Corresponding Effected Plane	Absolute Entropy Difference
256 × 256 × 1 = 65536	Plane 1	0.00379
256 × 256 × 2 = 131072	Plane 2	0.01531
256 × 256 × 3 = 196608	Plane 3	0.04353
256 × 256 × 4 = 262144	Plane 4	0.1544
256 × 256 × 5 = 327680	Plane 5	0.229
256 × 256 × 6 = 393216	Plane 6	0.3885
256 × 256 × 7 = 458752	Plane 7	0.8663
256 × 256 × 8 = 524288	Plane 8	0.8929

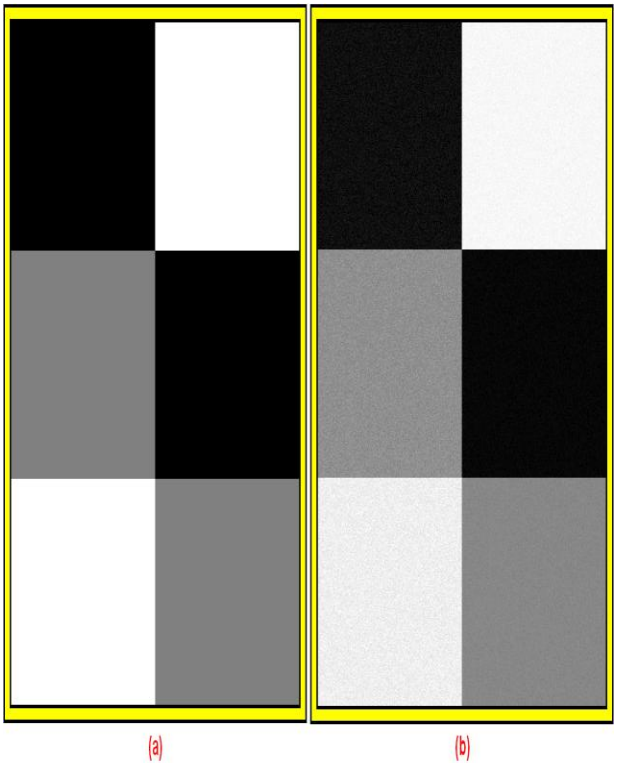

Fig 8: Original image (b) Degraded image by Bit Plane Splicing LSB technique due to partial data filling

The proposed algorithm is reversible as the secret data is recovered properly. Note that we are assuming the received image contains no noise or impairments due to transmission channel. In order to protect the stego image from channel induced interference, channel encoding and modulation techniques can be used but this is not the topic of this paper.

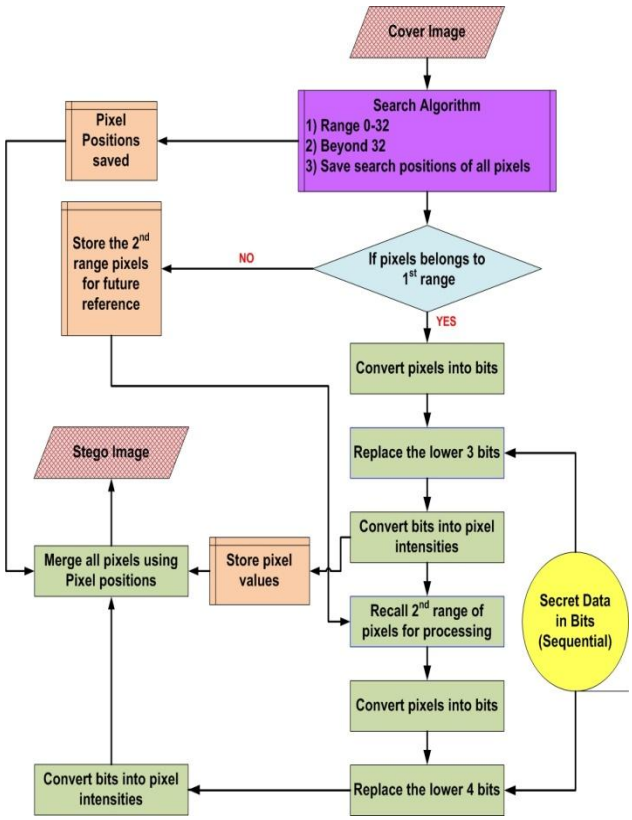

Fig 9: Flow diagram of the proposed OIBDH algorithm at the sending end

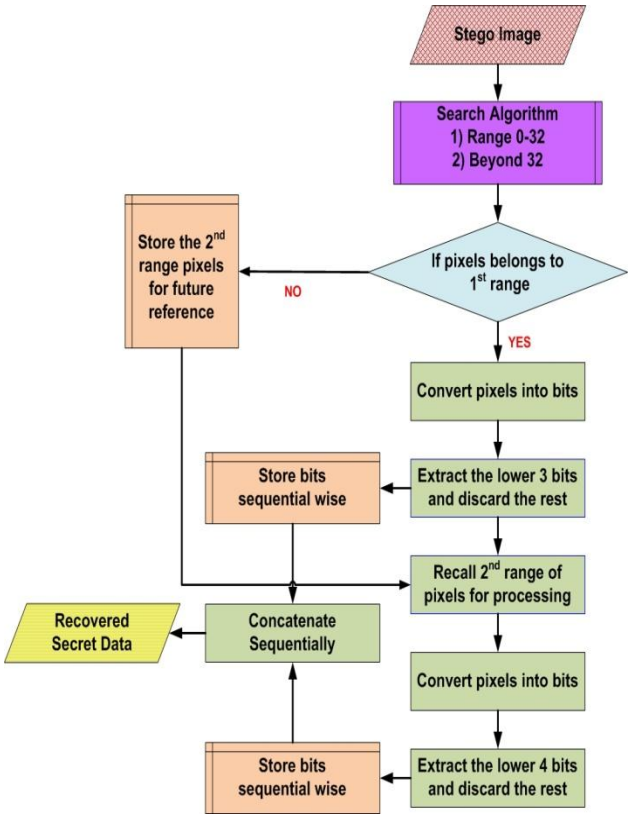

Fig 10: Flow diagram of the proposed OIBDH algorithm at the receiving end to recover the hidden secret data

Hence we are assuming a perfect channel in order to explain the recovery process. The recovery process as explained in Fig. 10 is same as the hiding process. Each pixel is inspected for its intensity value. If a pixel belongs to the first range, then the last three bits are extracted. This process continues till all the pixels in the first range are exhausted after which the second range of pixels are found and the lower four bits are extracted. It is clear that during hiding process and after inserting the secret bits, the intensity values of pixels change but the intensity range remains the same. In this way we do not need any extra step to recover our secret data because the intensity range for pixels remains the same for all pixels. Furthermore, to recover the secret data properly, the size of secret data should be transmitted or shared between the sender and receiver to make sure that false data is not extracted from those pixels which do not contain secret bits.

The proposed algorithm outperforms the conventional Bit Plane Splicing LSB technique as it generates less distortion in the stego image. This is further elaborated in the next section and test results are presented.

4. PERFORMANCE EVALUATION OF OIBDH ALGORITHM

To prove the superiority of OIBDH over Bit Plane Splicing LSB technique, a series of experiments are performed. In the first experiment both algorithms are applied on a test image containing all the gray scale shades between 0 and 255 shown in Fig. 11 and the same data is used for hiding in both algorithms.

As it is evident from Fig. 11 (b) and (c) the degradation is more severe using Bit Plane Splicing LSB technique than in OIBDH especially in the gray region near darker side. Also note how the transition of gray levels changes abruptly in Bit Plane Splicing LSB technique and less abruptly in OIBDH. This again supports our assertion regarding OIBDH algorithm. Furthermore, to compare the performance of both algorithms in terms of entropy curves, both algorithms are applied on camera man image with different secret data sizes as shown in Fig. 12. The entropy difference after each insertion is calculated in both situations. In addition, the experiment is repeated for 3 sets of images belonging to camera man image but with different variations in brightness (i.e. original, bright and dark). Note that the image size in bits for all three images is same i.e. ($300 \times 300 \times 8 = 72000$ bits) and the entropy of the original, bright and dark cover images are 7.2738, 6.5507 and 5.9121 respectively. Also note that the maximum secret data size or cover image capacity is different for both algorithms. In case of Bit Plane Splicing LSB technique the maximum capacity is same as the cover image size in all three cases (i.e. 72000 bits) where as in case of OIBDH and because of the hiding rules mentioned in the previous section, the maximum secret data sizes for hiding in the three images are 287689, 270000 and 298456 bits respectively. All the curves are plotted on log scale for better view and for better comparison. As we can see from the generated curves the absolute entropy difference curve in case of OIBDH is below the curve generated by Bit Plane Splicing LSB technique in all three sets. The comparison is done till the maximum capacity point

for OIBDH as data sizes beyond this point is not allowed in OIBDH algorithm as mentioned earlier. These curves show that for the same amount of data to be hidden, degradation measured by absolute entropy difference parameter in the stego image is more severe (i.e. greater absolute entropy difference values) in case of Bit Plane Splicing LSB technique than our proposed OIBDH algorithm. In addition the gap between the two curves increases as the image gets darker. This shows that darker images are better candidates to be used as cover images as less degradation occurs after data insertion. Although at some points absolute entropy difference of OIBDH is more than Bit Plane Splicing LSB but generally it is less.

Finally to have a visual look on the quality degradation of cover image for both algorithms, both algorithms are used for inserting 287689 bits into Lena's image (this is the maximum hiding capacity for OIBDH for Lena's image). It is clear that more degradation is obvious in Bit Plane Splicing LSB and less degradation in OIBDH as shown in Fig. 13. The artifact effect is visually very clear especially in uniform gray areas e.g. shoulder and cheeks indicated by yellow circles in Fig. 13 (b).

5. CONCLUSION

In this paper an algorithm called Optimum Intensity Based Distributed Hiding (OIBDH) is proposed for secret data hiding inside a cover image. This algorithm is an improved version of Bit Plane Splicing LSB technique. Secret data are not hidden sequential wise and plane by plane as in Bit Plane Splicing LSB technique, rather data are hidden in a non-sequential manner and not plane by plane. The amount of bits to be hidden in a pixel depends on the gray scale value of the pixel. This makes the proposed algorithm dynamic in nature and more effective both in terms of data hiding capacity and visual degradation of the cover image than Bit Plane Splicing LSB technique. Results in the form of absolute entropy difference curve and visual degradation of various cover images show the superiority of OIBDH over Bit Plane Splicing LSB technique.

6. REFERENCES

[1] Dapeng Wu, Yiwei Thomas Hou, Wenwu Zhu, Ya-Qin Zhang and Jon M. Peha, "Streaming Video over the Internet: Approaches and Directions", *IEEE Transactions on circuits and systems for video technology*, vol. 11, no. 3, pp. 282 – 300, March 2001.

[2] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich and T. Kalker, "Digital Watermarking and Steganography", *2nd Edition, Morgan Kaufmann Publishers*, 2008 (ISBN 978-0-12-372585-1).

[3] R.J. Anderson and F.A.P. Petitcolas, "On the Limits of Steganography", *J. Selected Areas in Comm.*, vol. 16, no. 4, pp. 474 – 481, May 1998.

[4] Sung, A.H., Tadiparthi G.R. and Mukkamala S., "Defeating the current steganalysis techniques (robust steganography)", *in proc. of The International Conference on Information Technology: Coding and Computing (ITCC 2004)*, Las Vegas, Nevada, USA, vol. 1, pp. 440 – 444, April 2004.

[5] Chiou-Ting Hsu and Ja-Ling Wu, "Hidden Digital Watermarks in Images", *IEEE Transactions on Image Processing*, vol. 8, no. 1, pp. 58 – 68, January 1999.

[6] Soumik Das, Pradosh Bandyopadhyay, Shauvik Paul, Atal Chaudhuri and Monalisa Banerjee, "An Invisible Color Watermarking Framework for Uncompressed Video Authentication", *International Journal of Computer Applications*, vol. 1, no. 11, pp. 22 – 28, February 2010.

[7] Chun-Hsiang Huang and Ja-Ling Wu, "Attacking visible watermarking schemes", *IEEE Transactions on Multimedia*, vol. 6, no. 1, pp. 16 – 30, Feb. 2004.

[8] Guorong Xuan, Jiang Zhu, Jidong Chen, Shi, Y.Q., Zhicheng Ni and Wei Su, "Distortionless data hiding based on integer wavelet transform", *Electronic letters*, vol. 38, no. 25, pp. 1646 – 1648, Dec. 2002.

[9] Faisal T. Alturkia, Ali F. Almutairib, and Russell M. Mersereauuc, "Analysis of blind data hiding using discrete cosine transform phase modulation", *Signal Processing: Image Communication*, Vol. 22 no. 4, pp. 347 – 362, April 2007.

[10] Chou, J. and Ramchandran K., "Robust turbo-based data hiding for image and video sources", *in proc. of IEEE International Conference on Multimedia and Expo 2002 (ICME '02)*, Lausanne, Switzerland, August 2002, vol. 2, pp. 565 – 568.

[11] Sandipan Dey, Ajith Abraham and Sugata Sanyal, "An LSB Data Hiding Technique Using Prime Numbers", *in proc. of the Third International Symposium on Information Assurance and Security (IAS '07)*, Manchester, United Kingdom, August 2007, pp. 101 – 106.

[12] Chi-Kwong Chan and L. M. Cheng, "Hiding Data in Images by Simple LSB Substitution", *Pattern Recognition*, vol. 37, no. 3, pp. 469 – 474, March 2004.

[13] Cheng-Hsing Yang, Chi-Yao Weng, Shiuh-Jeng Wang and Hung-Min Sun, "Adaptive Data Hiding in Edge Areas of Images With Spatial LSB Domain Systems", *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 488 – 497, Sep. 2008.

[14] Cheng-Hsing Yang,"Inverted pattern approach to improve image quality of information hiding by LSB substitution", *Pattern Recognition*, vol. 41, no. 8, pp. 2674 – 2683, Feb. 2008.

[15] Keiji Yanai and Kobus Barnard, "Image region entropy: a measure of "visualness" of web images associated with one concept", *in proc. of the 13th annual ACM international conference on Multimedia*, Singapore, November 2005, pp. 419 – 422.

[16] Norman Abramson, "Information Theory and Coding", *1st edition, McGraw-Hill Education*, 1963.

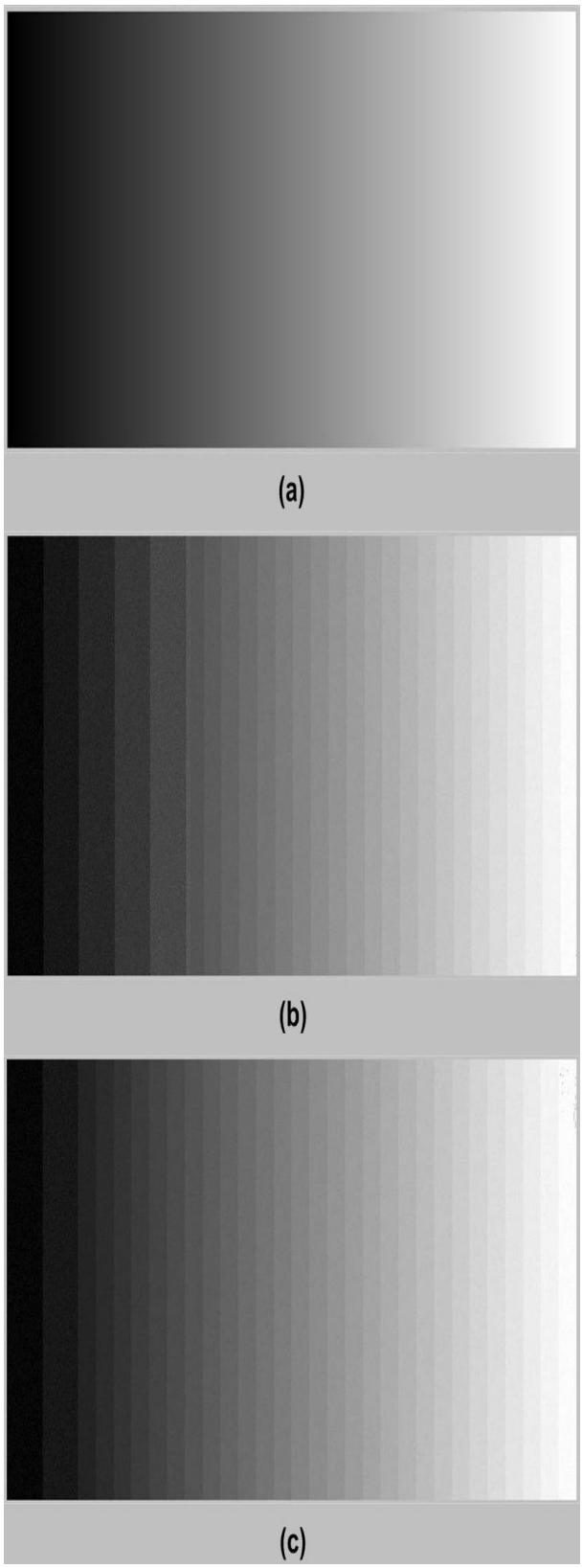

Fig 11: (a) Image degradation of gray intensity transition image when: (b) Bit Plane Splicing LSB technique is applied (c) OIBDH is applied

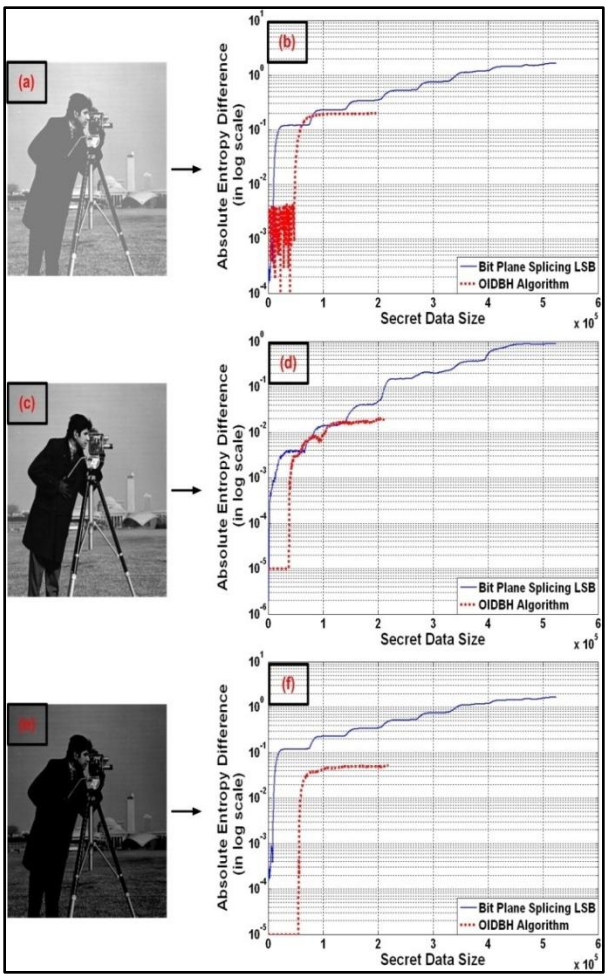

Fig 12: Degradation performance evaluation of Bit Plane Splicing LSB and OIBDH for camera man image with three brightness levels (a) bright (c) middle or original (e) dark

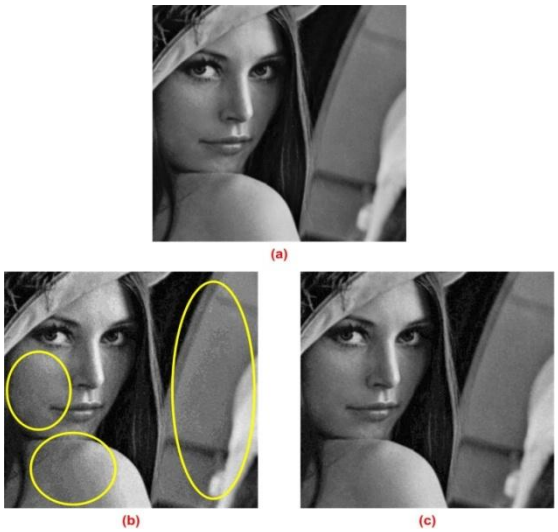

Fig 13: Artifact effect on cover image of (a) Lena when (b) Bit Plane Splicing LSB technique and (c) OIBDH algorithm are applied