ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2017.0214

HIDING TEXT IN DIGITAL IMAGES USING PERMUTATION ORDERING AND COMPACT KEY BASED DICTIONARY

Nagalinga Rajan[1] and R. Sunder[2]

[1]Department of Statistics, Manonmaniam Sundaranar University, India
[2]Department of Computer Science and Engineering, MET's School of Engineering, India

Abstract

Digital image steganography is an emerging technique in secure communication for the modern connected world. It protects the content of the message without arousing suspicion in a passive observer. A novel steganography method is presented to hide text in digital images. A compact dictionary is designed to efficiently communicate all types of secret messages. The sorting order of pixels in image blocks are chosen as the carrier of embedded information. The high correlation in image pixel values means reordering within image blocks do not cause high distortion. The image is divided into blocks and perturbed to create non repeating sequences of intensity values. These values are then sorted according to the message. At the receiver end, the message is read from the sorting order of the pixels in image blocks. Only those image blocks with standard deviation lesser than a given threshold are chosen for embedding to alleviate visual distortion. Information Security is provided by shuffling the dictionary according to a shared key. Experimental Results and Analysis show that the method is capable of hiding text with more than 4000 words in a 512×512 grayscale image with a peak signal to noise ratio above 40 decibels.

Keywords:

Digital Images, Steganography, Permutation Ordering, Lehmer Codes, Text Hiding

1. INTRODUCTION

Information security and secure communication are crucial to the emerging hyper connected world. Cryptography emerged as the means of secure communication after the Second World War. With the advent of cheaper computing devices, it has played a dominant role in information security. It relies on making the communication non-readable by intercepting observers. However the presence of secret messages is apparent in cryptographic communication. In certain scenarios, this can lead to the destruction of the medium by authoritative regimes intent on stopping the communication. The senders and receivers may be even deemed guilty until proven otherwise. This leads to the need of Steganography, which hides the secret messages under the pretext of legitimate communication [1].

There are many types of steganography depending on the cover data used. Images, Text, Audio and Video are some of the examples. Digital images are the ideal cover medium due to the prevalent use in social media and the presence of large redundancy in image data [2]. According to the Mary Meeker Internet Trends report 2016 [3], more than three billion digital images are uploaded in the internet on a daily basis as of 2015.

Digital Image Steganographic techniques are designed with three criteria, namely imperceptibility, capacity and effectiveness against analysis [4]. Imperceptibility refers to the inability of passive observers to visually distinguish a stego image from a normal image. It is measured in the Peak Signal to Noise Ratio between the cover and stego images. Capacity is the amount of data that can be hidden per pixel of the cover image. The stego image must be undetected by one of the many steganalytic techniques which are based on statistical analysis of pixel values.

Usually binary data is assumed in designing steganographic techniques. It is possible to specialize the embedding techniques with the knowledge of the nature of the messages. The ability to securely communicate text messages has huge potential. Unlike other encrypted binary data, the common text message in English language has limited vocabulary. This fact can be exploited by providing code symbols for the most commonly used words. This greatly enhances the embedding capacity of the steganographic schemes.

Hiding Data in digital images are not only used to pass secret messages but also is used for image authentication, image tampering detection and copyright enforcement etc. The methods that aid in hiding information in digital images for the aforesaid purposes are called watermarking methods. In visible watermarking, the embedded information is visible to the naked eye. Usually this is done to prevent copyright violation of image assets. Invisible watermarking is similar to steganography in that it hides the message but it is done for tamper detection, authentication etc. The embedding capacity is required to be fixed in watermarking while higher capacity is better for steganographic purposes.

Classical methods of secret communication relied on the secrecy of the agreed method of communication. A popular example is that of the Roman general who would write message on the shaved head of a slave. They waited till the hair grew back to cover the writing on the scalp and then send the slave to deliver the tattooed message. Once the secret method is known this system would fail because it is easy to check the scalp of every passing slave. But this type of secret communication does not scale well in modern situations when the communicating parties must quickly establish a secure channel without a lot of pre arrangement. As the method is publicly known, security is provided by the knowledge of secret keys. The secret keys are exchanged through specialized protocols to establish the secure channel. Exchanging keys through a steganographic medium is a challenging open research problem.

The common form of digital image steganography involves replacing the least significant bit (LSB) of the image pixel intensity values with the message bits as in [5]. Digital Images are represented as an array of 8 bit unsigned integers. Let P be the intensity value of a pixel. It takes values in the range from 0 to 255. For example, if $P = 123$ and the message digit is 0, then the last bit of P is replaced with 0 to get the new value of 122. This modification changes roughly half of the pixels and the maximum change is ±1. Several methods are proposed in the literature based on LSB embedding. They vary from one another in the number of bits, the choice of cover pixels and the choice of the carrier signal.

This paper presents a system that can steganographically transmit large amounts of text data through digital images. Related methods are discussed in section 2. The drawbacks of existing system are discussed in section 3. The proposed system is presented in section 4, results and discussion are presented in section 5 and conclusion is offered in section 6.

2. LITERATURE SURVEY

The steganographic techniques in literature fall under two class namely spatial domain and transform domain.

Transform domain methods apply one of the many image transforms and then operate on the coefficients. The image pixels are highly correlated but in the transform domain, the coefficients are less correlated. So this allows the choosing of the specific coefficients for the optimal performance. The choice of the coefficients is decided by the tradeoff between increasing the embedding capacity and decreasing the distortion. JSteg [6], F5 [7], YASS [8], DWT based steganography [9] etc. fall under this category.

Spatial domain methods operate on the intensity values of the pixels themselves or some properties of the pixels. LSB embedding is the classical example of this approach. Histogram modification techniques rely on shifting the histogram of pixels or the histogram of prediction errors. The LSBs of prediction errors are useful in hiding compressed image data.

LSB Matching Revisited [10] used a pair of pixels as an embedding unit to reduce distortion. Exploiting Modification Direction (EMD) method [11] reduced the distortion for the given embedding capacity by hiding bits in pairs of pixels instead of single pixels. Diamond Encoding improved the capacity which is further enhanced by Adaptive Pixel Pair Matching [12]. Adaptive Pixel Triplet Matching improved the effective capacity even more by using three pixels as an embedding unit [13].

Model Based methods utilize image models to control the distortion in a content adaptive fashion. Highly Undetectable Steganography (HUGO) [14], Universal Wavelet Relative Distribution (UNIWARD) [15], Gibbs Construction [16] etc. are examples of this approach. Spread spectrum steganography [17] techniques hide the stego signal by making it statistically and visually indistinguishable from genuine image noise. Several improvements have been proposed for Spread Spectrum techniques.

Edge Adaptive LSB Matching Revisited [18] was proposed to prefer LSBs of edge pixels for hiding the stego signal. This approach preserved the smooth areas of the image and moved the embedding to textured areas that are less sensitive to distortion.

Lehmer code based method [19] was proposed where the carrier signal is the sorting order of groups of pixels in coherent image blocks. Image blocks with high variance are avoided. The rest of the blocks are embedded with secret message digit by perturbing the pixel intensity values so that the sorting order represents the message digit in Lehmer code space. The usage of Lehmer code and the sorting order has high potential because the number of possible permutations is the factorial function which grows very quickly with the number of pixels in the block. Rearrangement of pixel in image pixels can cause high distortion where the pixels values have high variance. This method restricts the image blocks on the basis of a variance threshold to alleviate the distortion.

This paper is based on Lehmer Code Based Steganography (LCBS) specialized to hiding English language text. The contributions of this paper are twofold. First instead of using binary form of data, English words are directly mapped to permutation ordering. Secondly, a compact dictionary is proposed to be used whose word order is scrambled with a shared secret key. This mechanism takes care of the security aspect. The method does not rely either on the secrecy of the dictionary or the steps. Experimental results indicate that large amounts of text data can be transmitted through digital images. Security analysis is carried out to show that the proposed method cannot be detected through conventional steganalysis methods.

3. EXISTING SYSTEM

3.1 THEORETICAL BACKGROUND

The embedding procedure of LCBS is briefly presented and its drawbacks discussed in this section.

Let *C* be the cover grayscale image of height *h* and width *w*. The image is divided into blocks of size *s*×*s* without overlap. The 3×3 block centered around the pixel $C_{i,j}$ consists of the pixels given by

$$B_{i,j} = \left\{ \begin{matrix} C_{i-1,j-1} & C_{i-1,j} & C_{i-1,j+1} \\ C_{i,j-1} & C_{i,j} & C_{i,j+1} \\ C_{i+1,j-1} & C_{i+1,j} & C_{i+1,j+1} \end{matrix} \right\} \tag{1}$$

where, i ={1,2,..,h} and j = {1,2,..,w} are the row and column indices.

The pixel values in each of the blocks are highly correlated and remain within a small range. The blocks are tested to determine whether they can be perturbed and scrambled without causing too much distortion. The test consists of calculating the standard deviation of the values in the block. It is given for a 3×3 block by,

$$\sigma_{i,j} = \frac{\sum_{x=i-1}^{x=i+1} \sum_{y=j-1}^{y=j-1} \left(C_{x,y} - \mu_{i,j} \right)^2}{9} \tag{2}$$

$$\sigma_{i,j} = \frac{\sum_{x=i-1}^{x=i+1} \sum_{y=j-1}^{y=j-1} C_{x,y}}{9} \tag{3}$$

The test is given as:

$$T_{i,j} : \sigma_{i,j} < t \tag{4}$$

The blocks that pass the test are chosen for embedding and rest of the blocks are left untouched. The embedding procedure does not change the value of the standard deviation. Therefore the test returns the same set of blocks at both the sender and receiver end.

The values are then perturbed to remove any repetition. The procedure of perturbation used in this paper is described here. The values are sorted in ascending order. The repeating values are replaced with the closest unsigned 8 bit integer not present in the block. This step is needed to ensure that the block of pixels encode the permutation order. Then the message digit is encoded as a permutation order using Lehmer code and the block of pixels are rearranged in that order. The message digit can be one of the 362880 values in the range,

$$m_{i,j} = \{1,2,..,9!\} \tag{5}$$

Each of the digit in this range can be encoded as a permutation order of the set of values {1,2,3,4,5,6,7,8,9} using Lehmer codes. The final step is the rearranging of the order of pixels in row first manner to represent the permutation corresponding to the message digit. At the receiver end the sorting order if the pixels is read and converted to the message digit.

Lehmer code is a particular way to encode each possible permutation of a sequence of n numbers. It is an instance of a scheme for numbering permutations and is an example of an inversion table. The Lehmer code makes use of the fact that there are $n!$ permutations of a sequence of n numbers.

Let σ denote a particular permutation of a sequence of integers from 0 till n-1. A pair of indices (i,j) with $i < j$ and $\sigma_i > \sigma_j$ is called an inversion of σ and the Lehmer code $L(\sigma)_i$ counts the number of inversions with fixed i and varying j. The sum $L(\sigma)_1+L(\sigma)_2 +\cdots+ L(\sigma)_n$ is the total number of inversions of sigma which is also the number of adjacent transpositions that are needed to transform the permutation into the identity permutation. $L(\sigma)$ represents σ and can be used to encode the permutation.

To get the Lehmer code of a particular sequence the following in-place procedure is adopted. For every number of the sequence x starting with the first one, the numbers to the right of x which are greater than x are reduced by 1. The sequence thus obtained is expressed in factorial radix giving the integer encoding of the permutation. For example the permutation (1,0,3,4,2) undergoes the mentioned procedure.

$$(1,0,3,4,2)$$
$$(1,0,2,3,1)$$
$$(1,0,1,2,0)$$
$$(1,0,1,1,0)$$
$$(1,0,1,1,0)$$
$$L(1,0,3,4,2) = 1\times4! + 0\times3! + 1\times2! + 1\times1! + 0\times0! = 27$$

The procedure can be reversed to decode the permutation from the Lehmer code. LCBS exploits the high value of the factorial function to encode large amount of message bits by converting them to a higher base, 9! in this case.

3.2 DRAWBACKS OF EXISTING SYSTEM

LCBS converts message bits to a higher base. This allows for the embedding of $\log_2 9! \cong 18$ bits in a 3×3 block yielding an embedding capacity of 2 bits per pixel (bpp). However to encode a text symbol using ASCII code having 8 bits, the embedding capacity will be 0.25 characters per pixel (cpp). For a grayscale image of size 512×512 with half of the blocks usable, around 4kb text data can be embedded. This capacity may be insufficient for real world applications which require messages with attachments. Also the process of converting to the higher base is unnecessarily complicated and error prone. The capacity can be increased much more by embedding an entire word in a block. This is achieved by enumerating commonly used English words along with alphanumeric symbols and mapping them directly to permutation order by Lehmer codes. LCBS is not adapted to text data in particular and this work addresses that issue.

4. PROPOSED SYSTEM

4.1 COMPACT KEY BASED DICTIONARY

A Compact Key Based Dictionary is proposed for the purpose of communicating secret text messages through a steganographic medium. The carrier signal for the communication is chosen to be the sorting order of intensity values in 3×3 image pixel blocks. It is ensured that the nine values in a block are non-repeating. This provides a total number of 9! sorting orders which is 362880. Therefore the dictionary is designed with 362880 English words and Characters. Similar dictionaries can be designed in other languages with ease.

The Table.1 provides the list of word categories included in the dictionary. The ASCII characters are included so that any word not included in the dictionary can be spelled out. This includes both uppercase and lowercase alphabets, Arabic numerals, Space and other Special Characters like punctuation marks, line feed etc. These characters are used only when the complete word is not part of the dictionary. This is done to ensure that maximum amount of text information may be hidden in the given digital image. Words that indicate time in HH:MM format for every complete five minutes like 00:00, 00:05, 00:10 are included and two words for “AM” and “PM” are included. This allows the communication of time information compactly. Peter Norvig's 1/3 Million Frequent Words list and 1/4 Million two word bigrams list are used and the top 3,00,000 words and 33,000 bigrams are included in the dictionary. This word list [20] is based on the Google Trillion Words Project [21] which lists the n-grams from millions of text material in Google's corpus. The remaining 29478 positions are left freely available for the communicating parties to customize for their special needs. They can publish this updated list without fear of compromising the communication itself. This customization is purely optional and all types of secret communication can be made without these additional words. Thus the dictionary is over complete and is fully adequate for compact communication.

Table.1. Components of the Compact Key Based Dictionary

Word Category	Description	No. of Words
ASCII Characters	These are used to create words and nouns not present in the dictionary.	256
Time in HH:MM AM/PM format	These are used to indicate time and includes every five minutes in a day. The words “AM” and “PM” take up two word positions. There are 144 time points every five minutes.	146
Common Words in English	Most commonly used English words in Peter Norvig's 1/3 Million Frequent words list.	300000
Common Bigrams in English	Most commonly used English words in Peter Norvig's 1/4 Million Frequent two word bigrams list.	33000
Additional Words	Additional word slots are used to customize the dictionary according to the requirements of the communicating parties.	29478

The content of the dictionary is made publicly available and can be easily shared by the sender and the receiver. But the ordering of the words depends on a shared Key. This key is a secret shared key between the sender and the receiver. It is assumed that the list of words in the dictionary is known to everyone including sender, receiver and authorized and unauthorized observers. The security of the system is based on the ordering of the words in the dictionary and not on its contents. Before using the dictionary, the dictionary is shuffled according to the shared key. This is done by generating a random permutation order using the key as the seed of the random generator. An efficient implementation of generating random permutation order used in this work is given by the *randperm* function in MATLAB software. Each of the words are mapped to random permutation of the set {1,2,3,4,5,6,7,8,9} using Lehmer Codes. So the embedding is done by matching the intensity value ordering in the image pixel blocks to the words in the secret message.

4.2 EMBEDDING PROCEDURE

The embedding procedure is as follows:

Step 1: The given cover image is divided into blocks of size 3×3.

Step 2: The standard deviation $\sigma_{i,j}$ of the values in each block is calculated and the blocks with $\sigma_{i,j}$ < t are selected for embedding.

Step 3: The selected block values are perturbed to have unique values. They are sorted first and the repeating values are increased or decreased by the smallest number in the set {1,2,3,4,5,6,7,8,9} to make them unique.

Step 4: The compact dictionary is shuffled using the shared key *k* as the seed for the pseudo random generator.

Step 5: The message text is converted to a series of integers in the range [1,362880] by taking the indices in the shuffled dictionary. Customized Additional words are detected first, followed by Time, Bigrams, One grams and finally ASCII characters.

Step 6: The message indices are converted to permutation of the set {1,2,3,4,5,6,7,8,9}.

Step 7: The chosen perturbed blocks are permuted according to the message indices one by one.

Step 8: The permuted blocks are arranged in the same place to get the stego image.

The block diagram for the embedding procedure is given in Fig.1. The perturbation increases the standard deviation by a little bit so the same blocks will be chosen at the receiver end for message extraction.

4.3 EXTRACTION PROCEDURE

The extraction procedure is as follows:

Step 1: The given stego image is divided into blocks of size 3×3.

Step 2: The standard deviation $\sigma_{i,j}$ of the values in each block is calculated and the blocks with $\sigma_{i,j} < t$ are selected for extraction.

Step 3: The sorting order of the selected blocks are converted to indices using Lehmer Code algorithm.

Step 4: The compact dictionary is shuffled using the shared key *k* as the seed for the pseudo random generator.

Step 5: The indices are converted to word, bigrams etc. using the shuffled dictionary to get the secret message.

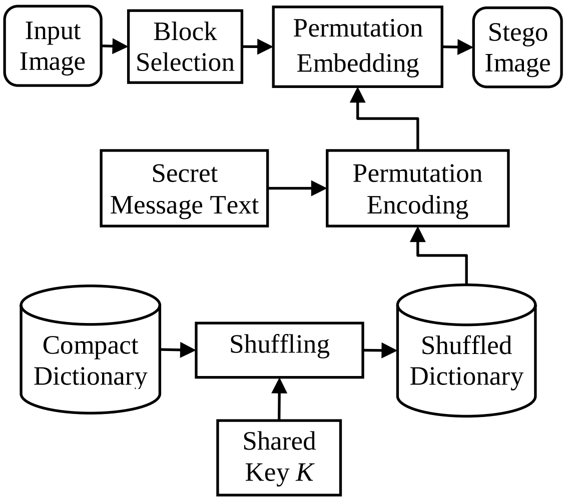


Fig.1. Block Diagram of the Embedding Procedure

The block diagram for the extraction procedure is given in Fig.2.

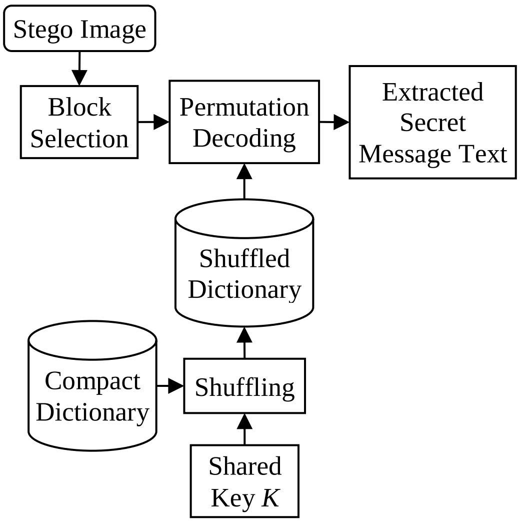


Fig.2. Block Diagram of the Extraction Procedure

5. RESULTS AND DISCUSSION

5.1 EXPERIMENTAL RESULTS

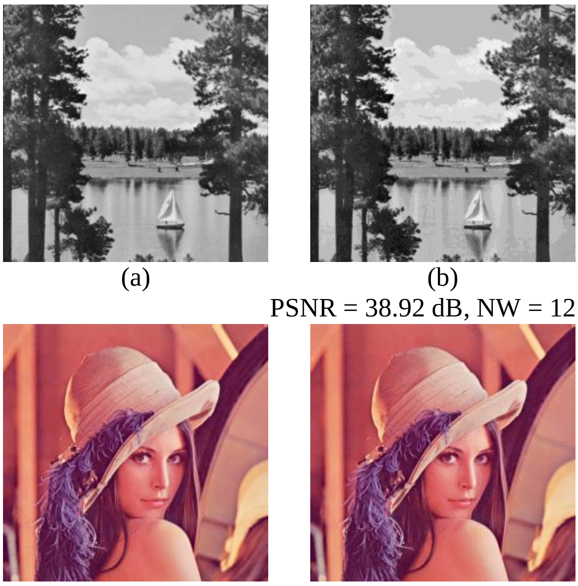
The proposed method is implemented in MATLAB 2013. Experiments are conducted using a test image dataset taken from USC-SIPI database and popular test images used in image processing research. Four images are used for illustration purposes namely Lake, Lena, Pirate and Walkbridge. The Lena image is an example of color image in the dataset. The proposed method extends to color images without any modifications by treating the color planes as separate grayscale images. Overall around 100 images were used and all the results reported are averaged over the entire dataset.

The Fig.3 shows the cover and stego images using the proposed method. The stego images are captioned with the Peak Signal to Noise Ratio (PSNR) to demonstrate the visual imperceptibility. The PSNR for the cover image *C* and stego image *S* is defined using the Eq.(6) and Eq.(7) where both the images are of size *m*×*n*. The value 255 is the peak signal since 8 bit images are considered in this work.

$$PSNR = 10\log 10\left(\frac{255^2}{MSE}\right) \tag{6}$$

$$MSE = \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{\left(c_{i,j} - s_{i,j}\right)^2}{m \times n} \tag{7}$$

The number of blocks *NW* used for embedding or the number of embedding linguistic units (words, bigrams, ascii characters etc.) are also displayed along with the stego images in Fig.3.



(a) (b)

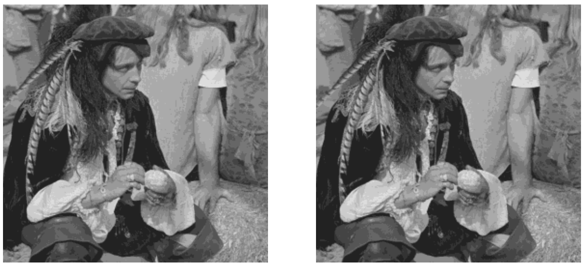
PSNR = 38.92 dB, NW = 12157

(c) (d)

PSNR = 38.48 dB, NW = 34088

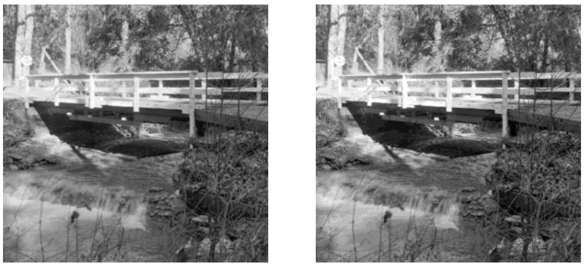
Fig.3. Cover and Stego Images Compared. a) Lake b) Stego Lake c) Lena d) Stego Lena

The average number of usable blocks or embedding units in the entire dataset was around 12000 for each of the 512×512 grayscale images and around 35000 for the color images. The Walkbridge is an example of a highly textured image for which the number of usable blocks are only around 4700. The high texture blocks are filtered out by the block selection criteria to avoid high distortion in the stego image. The visual imperceptibility is clearly demonstrated by the illustrated images in Fig.3 and Fig.4. The PSNR values are around 38 decibels for the chosen block size of 3×3 indicating good performance.



(e) (f)

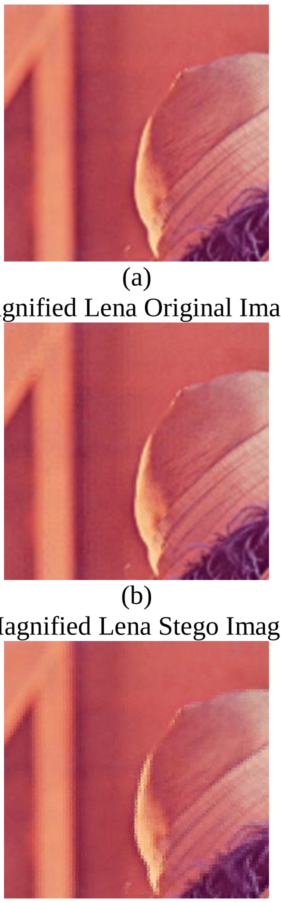
PSNR = 38.94 dB, NW = 13476



(g) (h)

PSNR = 42.52 dB, NW = 4703

Fig.4. e) Pirate f) Stego Pirate g) Walkbridge h) Stego Walkbridge



(a)

Magnified Lena Original Image

(b)

Magnified Lena Stego Image

(c)

Magnified Lena Stego Image without block Selection

Fig.4. Zoom-in views of Lena cover and stego images

The zoom-in views of the Lena cover and stego images are displayed in Fig.5 to show the visual clarity of the proposed method. The distortion produced by the permutation is not easily detectable.

If the textured blocks are selected for embedding the performance falls to 20 decibels or below. It also distorts the edges present in the image and produces many misaligned edges compromising the visual imperceptibility. So the block selection method is effective as shown by Fig.4. In this work the threshold parameter *t* is chosen as 5. The famous essays used as message texts used are shown in Table.2.

Table.2. Secret Message Texts Used in the Experiments

Title	Author	Size (KB)
"The Stage-Coachmen Of England: A Bully Served Out"	Borrow	23
"Ramblings In Cheapside"	Butler	31
"On History"	Carlyle	28
"My Winter Garden"	Kingsley	55
"Machiavelli"	Macaulay	95

5.2 PERFORMANCE ANALYSIS

The Fig.5 and Fig.6 illustrate the performance in terms of PSNR and embedding capacity in terms of number of words for the test images. The standard deviation threshold t is varied.

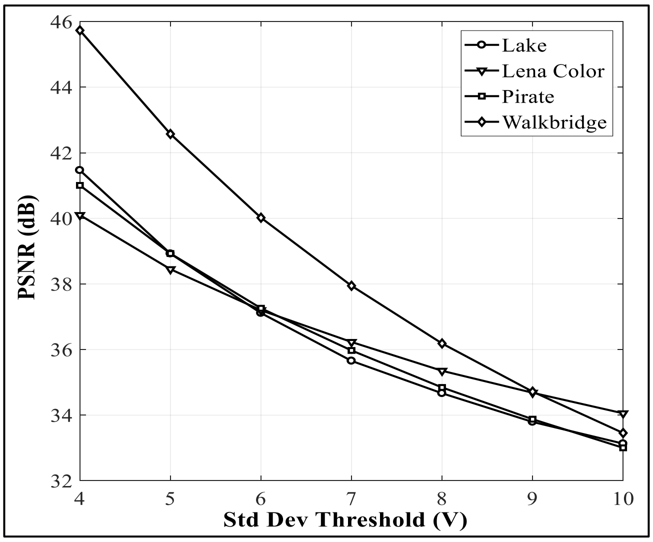


Fig.5. PSNR vs. Standard Deviation Threshold *t* for the illustrated images

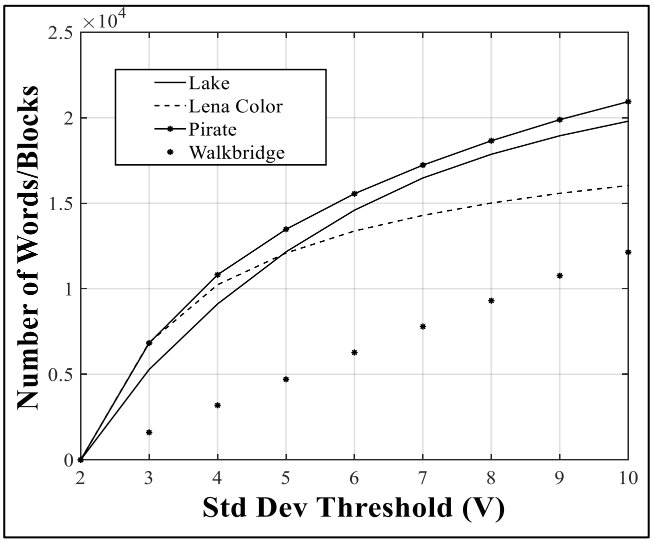


Fig.6. Embedding Capacity (Number of Words or Blocks) vs. Standard Deviation Threshold *t* for the illustrated images

The Fig.5 shows that for threshold value of 5, the expected PSNR performance is about 38 dB, which is sufficient for the purpose. For the choice of *t*=5, more than 12,000 blocks are available for embedding. With the use of compact dictionary, secret text messages spanning more than 10000 words can be embedded in a single image. Experimental results indicate that an average embedding capacity of 300 KB per image or around 9.4 bpp.

5.3 COMPARATIVE STUDIES

This section compares the performance of the proposed method against other steganographic methods. Most steganographic methods have an embedding capacity of less than 3 bpp. But the proposed method has high embedding capacities of more than 4 bpp. The Table.3 shows the improvement in embedding capacity of the proposed method compared to LCBS.

Table.3. Comparison of embedding capacities of proposed method against LCBS

Image	Embedding Capacity of LCBS (KB)	Embedding Capacity of Proposed Method (KB)
Lake	11.87	53.42
Lena	33.29	135.21
Pirate	13.16	55.27
Walkbridge	4.59	18.36

The performance of the proposed method is compared against LSB Matching method with 4 bpp for a fair comparison. The Table.4 shows the superior performance of the proposed method.

Table.4. Comparison of PSNR of proposed method against LSB Matching (4 bpp)

Image	PSNR of LSB Matching (dB)	PSNR of Proposed Method (dB)
Lake	25.40	38.92
Lena	29.06	38.48
Pirate	27.12	38.94
Walkbridge	31.01	42.52

5.4 SECURITY ANALYSIS

The proposed method makes very few changes in the pixel intensity values and only rearranges them within each block. So the histogram of the image remains largely untouched. The Fig.7 shows the normalized histograms for cover and stego lake image. The difference is shown in Fig.8. Since many steganalysis techniques rely on histogram statistics based attacks, the proposed method is highly effective against steganalysis.

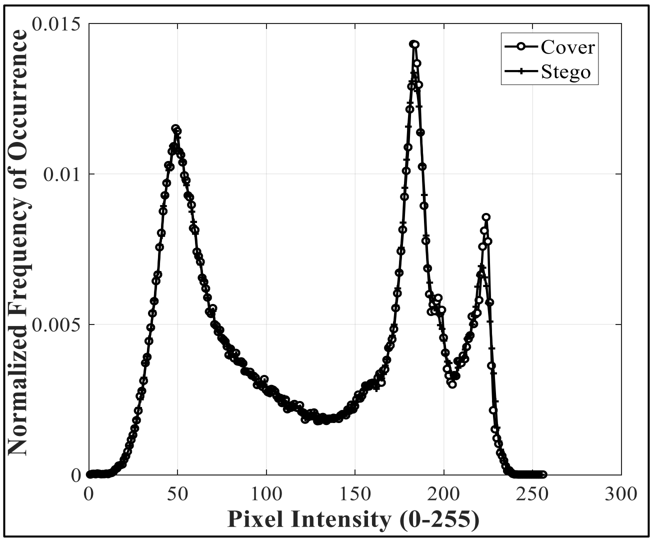


Fig.7. Normalized Histograms of Cover and Stego Lake Image

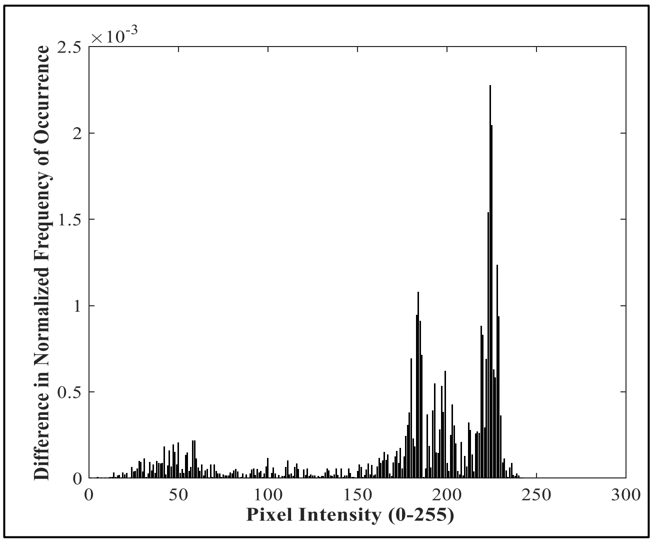


Fig.8. Difference in Normalized Histograms for Cover and Stego Lake Image

The Table.5 shows the average accuracy of common steganalysis techniques averaged over the entire dataset. The steganalysis techniques considered are ADJ-HCF-COM [22], RS [23], AUMP [24], SPAM [25], Shi 78-D [26], Farid 72-D [27], Moulin 156-D [28], and Li 110-D [29].

From Table.5, it can be clearly seen that the proposed method evades reliable detection by several steganalysis methods and therefore highly effective for secret communication. The effectiveness is as good as or better than edge adaptive LSB embedding which has comparatively lower embedding capacity.

Table.5. Average Accuracy (%) of Steganalysis Algorithms

	LSB Based	**Edge Based**	**Proposed Method**
ADJ-HCF-COM	73.21	51.01	56.21
RS Analysis	77.12	52.14	59.26
AUMP	79.56	59.84	56.51
SPAM	81.21	58.42	57.21
Shi 78-D	69.39	52.56	52.44
Farid 72-D	55.04	50.56	51.16
Moulin 156-D	57.61	51.39	52.26
Li 110-D	74.11	52.94	51.24
Max Accuracy	81.21	52.94	51.24

5.5 COMPUTATIONAL EFFICIENCY

The running times of the proposed method are shown in Table.6. MATLAB 2013 software running on Core i5 system with 4GB RAM is used as the benchmark.

Table.6. Computational Time taken for the proposed embedding algorithm

Image	**Computation Time (s)**
Lake	3.2
Lena	6.1
Pirate	3.1
Walkbridge	1.2

6. CONCLUSION AND FUTURE WORK

This paper presented a novel text data hiding technique in digital images. It utilized the Lehmer codes for embedding messages in the sorting order of pixel values in blocks. Experimental results show that the proposed method outperforms existing LSB based techniques, improves the embedding capacity than LCBS and more effective against steganalysis by preserving histogram characteristics. It delivers very high embedding capacities of more than 4.5bpp with high visual quality of around 38dB. Thus it is ideally suited for steganographic communication of large amounts of text data.

Future research can be directed in hiding other types of data such as binary, grayscale images or audio. The robustness of the method against common image processing modifications can also be improved by exploring transform domain processing.

REFERENCES

[1] K. Sayood, "Introduction to Data Compression", Available at:http://rahilshaikh.weebly.com/uploads/1/1/6/3/11635894/data_compression.pdf.

[2] Kamrul Hasan Talukderi and Koichi Harada, "Haar Wavelet Based Approach for Image Compression and Quality Assessment of Compressed Image", *International Journal of Applied Mathematics*, Vol. 36, pp. 1-9, 2007.

[3] Peter Wayner, "*Compression Algorithm for Real Programmer*", 1st Edition, Elsevier, 1999.

[4] Markus Kuhn, "Digital Signal Processing", Available at: https://www.cl.cam.ac.uk/teaching/0809/DSP/.

[5] Ken Cabeen and Peter Gent, "Image Compression and the Discrete Cosine Transform", Available at: https://www.math.cuhk.edu.hk/~lmlui/dct.pdf.

[6] Sean Dunn, Available at: http://davis.wpi.edu/~matt/courses/color

[7] Colm Mulcahy, "Image Compression using the HAAR Wavelet Transform", *Spelman Science and Math Journal*, pp. 22-31, 1997.

[8] Ryuji Matsuoka, Mitsuo Sone, Kiyonari Fukue, Kohei Cho and Haruhisa Shimoda, "Quantitative Analysis of Image Quality of Lossy Compression Images", Available at: https://pdfs.semanticscholar.org/e929/fe4e037d80a226549054fd35bced632e2009.pdf.

[9] James Z. Wang, "Wavelets and Imaging Informatics: A Review of the Literature", *Journal of Biomedical Informatics*, Vol. 34, No. 2, pp. 129-141, 2001.

[10] Yves Meyer, "Wavelets Algorithms and Applications", *SIAM Journal on Applied Mathematics*, Vol. 53, No. 1, pp. 1-6, 1993.

[11] Charles K. Chui, "*An Introduction to Wavelets*", Academic Press, 1992.

[12] C.K. Chui, "*Wavelets: A Tutorial in Theory and Applications*", Academic Press, 1992.

[13] Subhasis Saha, "Image Compression - from DCT to Wavelets: A Review", *Crossroads*, Vol. 6, No. 3, pp. 12-21, 2000.

[14] Li Wern Chew, Li-Minn Ang and Kah Phooi Seng, "Survey of Image Compression Algorithms in Wireless Sensor Networks", *Proceedings of International Symposium on Information Technology*, pp. 1-9, 2008.

[15] D. Cruz, T. Ebrahimi, J. Askelof, M. Larsson and C. Christopoulos, "Coding of Still Picture", *Proceedings of* 45th *SPIE Applications of Digital Image Processing*, Vol. 4115, pp. 1-10, 2000.

[16] Suchitra Shrestha and Khan Wahid, Hybrid DWT-DCT Algorithm for Biomedical Image and Video Compression Applications, *Proceedings of 10th IEEE International Conference on Information Sciences, Signal Processing and their Applications*, pp. 280-283, 2010.

[17] Swapna Subudhiray and Abhishek Kr. Srivastav, "Implementation of Hybrid Dwt-Dct Algorithm For Image Compression: A Review", *International Journal of Research in Engineering and Applied Sciences*, Vol. 2, No. 2, pp. 1200-1210, 2012.

[18] M. Shwetha, P. Ashwini and B.M. Sujatha, "Analysis of Image Compression Algorithms in WSN: A Review", *International Journal of Science, Engineering and Technology Research*, Vol. 3, No. 4, pp.1029-1032, 2014.

[19] Sonja Grgic, Kresimir Kers and Mislav Grgic, "Image Compression using Wavelets", *Proceedings of IEEE International Symposium on Industrial Electronics*, pp. 99-104, 1999.

[20] R.Sudhakar, R Karthiga and S. Jayaraman, "Image Compression using Coding of Wavelet Coefficients-A Survey", *ICGST-GVIP Journal*, Vol. 5, No. 6, pp. 25-38, 2005.

[21] Renu Sharma and Matish Garg, "Comparative analysis of JPEG DCT, Haar and Daubechies Wavelet, Fractal for Image Compression", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 4, No. 1, pp. 1227-1234, 2014.

[22] S. Sridhar, P. Rajesh Kumar and K.V. Ramanaiah, "Wavelet Transform Techniques for Image Compression-An Evaluation", *International Journal of Image, Graphics and Signal Processing*, Vol. 2, pp. 54-67, 2014.

[23] USC-SIPI Image Database, Available at: http://sipi.usc.edu/database/database.php?volume=misc, Accessed on 2014.