Vibrant Color Image Steganography using Channel Differences and Secret Data Distribution

MOHAMMAD TANVIR PARVEZ[1] AND ADNAN ABDUL-AZIZ GUTUB[2]

[1]*Department of Information and Computer Science, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia.*
Email: tparvez@kfupm.edu.sa

[2]Center of Excellence in Hajj and Omrah Research, Umm Al-Qura University, Makkah, *Saudi Arabia.*
Associate Researcher, Center Of Excellence in Information Assurance (CoEIA), King Saud University, Riyadh, *Saudi Arabia.*
Email: aagutub@ uqu.edu.sa

ABSTRACT

This paper presents a new algorithm for color image based steganography. Our algorithm adaptively determines the number of secret message bits that each pixel of cover image can store based on a partition scheme of the range of color intensity. This vibrant scheme is found to offer high capacity and security for the cover media. We also introduce a new approach for spreading the secret message over the whole cover image to reduce distortion in the resulting stego image. This data-spreading technique can be extremely effective in improving the security of the stego-system. The proposed method adaptively selects the best partition scheme of color values to spread the message most and increase security. Experimental analysis shows the effectiveness of our algorithm with very attractive results.

Keywords: Data Hiding; Data Spreading; Image Steganography.

INTRODUCTION

Steganography deals with embedding information in a given media (called cover media) without making any visible changes to it (Provos and Honeyman 2003). The goal is to hide a message within the cover media such that the existence of the message is concealed. Steganography research uses different cover media such as texts, sounds, video clips and images. In this paper, we consider color images as the cover media. There are many applications of steganography (Anderson and Petitcolas 1998, Cox et al. 2007), like hiding copyright information, avoiding snooping while

communication, data encapsulation (e.g. explanatory information within X-ray images), copyright protections, digital watermarking etc.

From algorithmic point of view, any steganography algorithm aims to achieve two goals: high security and high capacity (Parvez and Gutub 2008). High security means to have less distortion, both visual and statistical, and to make the data storage locations in the stego file not guessable. High capacity means that the algorithm should ensure a minimum capacity for a particular cover image and should be able to hide as much information as possible without increasing the distortion. In this paper, we introduce a novel algorithm for image based steganography which achieves both of these goals. High capacity is achieved by adaptively selecting the number of bits to store in a pixel of the cover image. We also introduce a very effective idea for reducing distortion in the stego image: data spreading.

There are a number of algorithms for image based steganography (Gutub et al. 2008, Chang et al. 2006, Chen et al. 2009, Chen 2007, Chen 2008, Li et al. 2006, Liu and Liao 2008, Noda et al. 2008, Sur et al. 2009, Wu et al. 2010, Yu et al. 2008, Yu et al. 2005). Johnson et al. (1998) discusses three popular methods for image steganography: LSB insertion, masking and filtering and algorithmic transformations. *LSB insertion* is a simple approach but vulnerable to even a slight image manipulation. *Masking and Filtering* hides information by marking an image, in a manner similar to paper watermarks. *Algorithmic Transformation* techniques like redundant pattern encoding, encrypt and scatter etc. exist which use different approaches for concealing messages. In *redundant pattern encoding*, a small message may be painted many times over an image so that if the stego-image is cropped, there is a high probability that the watermark can still be read. In *encrypt and scatter* the data are hidden throughout an image. Scattering the message makes it appear more like noise. Bailey and Curran (2006) evaluates seven different image based steganography methods namely Stego1bit, Stego2bits, Stego3bits, Stego4bits, StegoColourCycle, StegoPRNG, StegoFridrich.

Gutub et al. (2008) describes the pixel indicator technique. It uses the two least significant bits of one of the channels from Red, Green or Blue, as an indicator for existence of data in the other channels. The indicator channels are chosen in sequence, with R being the first. The disadvantage of this algorithm is that the capacity depends on the indicator bits. Moreover, the algorithm uses a fixed number of bits per channel to store data.

There are also a number of image based steganography products, like EzStego, J-Steg, F5 and S-tools. Most of these tools use JPEG images as the cover media since JPEG images requires less transmission bandwidth and less storage space. However, using uncompressed images (like BMP) offer potentially higher visual redundancy and may have larger capacity. In addition, the advent of high speed Internet overcomes the problem of larger transmission bandwidth required for uncompressed images. Therefore, in this paper, we focus on bitmap images as the cover media.

The proposed algorithm falls in the category of LSB replacement algorithms. The main disadvantage of the LSB replacement algorithm is the vulnerability against adversary attack and *steganalysis*, like the *chi-square* attack (Westfeld and Pfitzmann 2000). Our algorithm utilizes several novel approaches to overcome the vulnerabilities of LSB algorithms. The proposed method uses color intensity values of the different channels of a pixel to decide the number of bits to store in that pixel, rather than storing a fixed number of bits per pixel. Moreover, our algorithm dynamically spreads out the secret message throughout the cover image as much as possible to obtain visual imperceptibility, which in turn ensures higher security.

The proposed method utilizes the idea of a partition scheme (Parvez and Gutub 2008) of color intensity to determine number of bits to store in a pixel. However, the method by Parvez and Gutub (2008) is limited in a sense that the sender and receiver must agree on a partition scheme and it utilizes only a portion of the cover image to hide the secret message. The proposed algorithm adaptively selects the best partition scheme for a given cover image and secret data. It uses a statistical method to estimate the capacity of the cover media and determines the best partition scheme. The selected scheme ensures that the secret data is spread across the cover image and reduces distortion. Tests and experimental results show that our algorithm performs much better compared to the existing algorithms.

The rest of the paper is organized as follows. Section 2 describes the proposed algorithm. Section 3 deals with the experimental results. Finally, section 4 is the conclusion.

PROPOSED ALGORITHM

In this section, we describe our novel algorithm for color image based steganography. We describe our method for RGB images (bitmaps). However, the algorithm can be adapted to other color images as well.

Partition Scheme Approach

We first briefly describe the use of partition scheme to adaptively select the number of bits to be stored in a pixel. We call this algorithm as the *basic algorithm*.

A *partition scheme* is defined as a monotonically decreasing sequence $[a_i]$, where i = 1 to 8. Assume that the color intensity of a channel is c. Then, that channel with value c stores i number of data bits if $c >= a_i$ and for all j, where $j < i$, $c < a_j$. For correctness of the algorithm, we only use valid partitions schemes. We define a *valid* partition scheme as follows: let $[a_i]$ be a partition scheme where lower i bits of a_i is all 0's. Let $[b_i]$, with i = 1 to 8, be another sequence, where b_i is generated by setting the lower i bits of a_i all to 1's. If $a_i > b_{i+1}$, and i = 1 to 7, then $[a_i]$ is a *valid* partition scheme. This simple condition ensures that the same numbers of data bits are read from a channel in the receiver's side as stored in the sender's side. This is explained as follows.

Let $[a_i]$ be a valid partition scheme. Assume that a channel's intensity value $c >= a_i$ and for all j, $j < i$, $c < a_j$. Assume that the lower i bits of that channel are all changed to 1 (the extreme condition). Let the resulting intensity value for this channel be c'. The condition for valid partition scheme ensures that $c' >= a_i$ and for all j, $j < i$, $c' < a_j$. Thus the correct number of bits will be retrieved at the receiver's side.

Using a partition scheme to store secret data in a cover media is summarized as follows:

- Use one of the three channels as the indicator. The indicator sequence can be made random, based on a shared key between sender and receiver.
- Data is stored in one of the two channels other than the indicator. The channel, whose color value is lowest among the two channels other than the indicator, will store the data in its least significant bits.
- Instead of storing a fixed number of data-bits per channel, variable number of bits are to be stored depending on the color value of the channel. The lower the value, the higher the data-bits to be stored. Therefore a partition of the color-values is needed. Through experimentations, we show that optimal partition may depend on the actual cover image used.
- To retrieve the data, we need to know which channel stores the data-bits. This is done by looking at the least significant bits of the two channels other than the indicator:

- If the bits are same, then the channel following the indicator in cyclic order stores the data.
- Otherwise, the channel which precedes the indicator in cyclic order stores the data.

Here, the cyclic order is assumed to be R-G-B-R-G-B and so on. The appropriate bits can be set while the data is stored. Note that, it is assumed that a shared key is already agreed upon by the two parties and a valid partition scheme is selected.

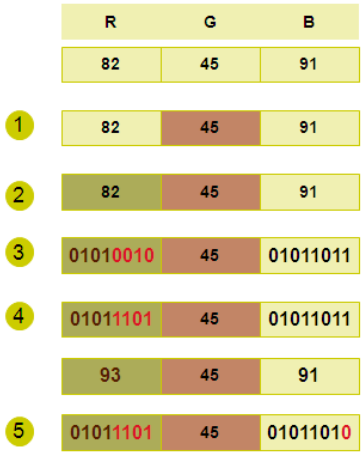


Figure 1: Illustration of hiding data bits inside a pixel.

Figure 1 demonstrates one example of storing data bits in a channel. The three channels R, G, and B contain values 82, 45, and 91. In step – 1, the indicator channel (here G) is known from the indicator sequence generated from a shared key. In step – 2, the data channel is chosen. Since channel R has the lower value among channels R and B, channel R is chosen to store data. In step – 3, number of data bits to store in the lower bits of channel R is determined from the current channel R value (here 82) and the agreed partition scheme. In the example, four lower order bits of channel R are used to store data. Step – 4 stores the data bits in the lower four bits of channel R. After we store the data, the lower four bits of channel R change from 0010 to 1101. This change depends on the actual data value being stored. After storing the data bits, the value for the channel R changes to 93, which is now greater than the value of channel B (91). This will create confusion while retrieving the data, if the receiver uses the same rule as the sender for selecting the channel to store data. Therefore, to retrieve the data correctly, the LSB of channel B is modified to 0, so that the LSBs of channels R and B do not match. While retrieving, the receiver checks the LSBs of

channels R and B and selects R, since the LSBs of channel R and B don't match and R precedes the indicator channel G in the cyclic sequence of channels.

Data Spreading using Adaptive Partition Schemes

The basic algorithm described above guarantees a minimum capacity for every cover image depending on the color intensity within the channel to hold secret data. The basic algorithm works only after a partition scheme is selected. In this section, we describe the method to select the best partition scheme for a cover image. Our method effectively selects a partition scheme that adaptively spread the secret across the cover image to obtain less distortion.

The selection of the best partition scheme is accomplished as follows:

- Define multiple partition schemes, each allowing different maximum number of bits to be stored per channel. For example, one partition scheme may allow changing up to 5 bits per channel, whereas another scheme may allow storing only 2 bits of data per channel.
- The partition scheme is chosen adaptively based on the cover image and the data file. No pre-agreed partition schemes between sender and receiver.
- Since different partition schemes would use different fractions of the cover image to store data, the partition scheme which uses *maximum* number of cover image pixels will be selected.
- Since partition scheme is chosen adaptively, the stego file will contain enough information to retrieve the partition scheme used. The first few pixels of the cover image will store the partition scheme.

Figure 2 shows the flow charts of the encoding and decoding steps when a partition scheme is adaptively selected.

An important part of our algorithm is the statistical estimation of the maximum number of bits that can be stored for a particular partition scheme and a given cover image. Assume that the partition scheme is give by $[a_i]$, where i = 1 to 8; a pixel in the cover image can take any value from 0 to 255 with uniform probability. We estimate the maximum capacity of that cover image for this partition scheme by calculating the ratio f_i, which represents the fraction of the range of values for which lower i bits of the cover image channel will be modified by the data bits. The estimation of f_i from a_i is performed using the following formula:

$$f_1 = \frac{256 - a_1}{256}$$

$$f_i = \frac{a_{i-1} - a_i}{256}, \quad i = 2 \ldots 8$$

Then, the maximum number of bits, *c*, which can be stored in the cover image using this particular partition scheme, can be estimated using the following formula:

$$c = \left\lfloor \sum_{i=1}^{8} i f_i S \right\rfloor$$

Here *S* is the number of pixels in the cover image.

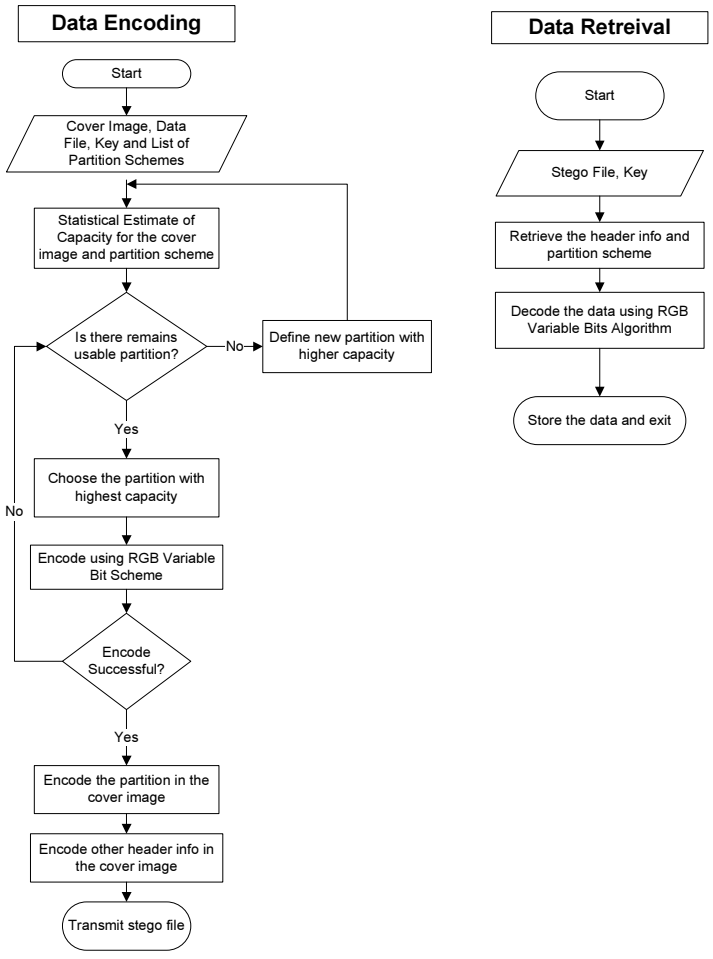


Figure 2: Encryption and decryption steps when a partition scheme is selected adaptively; data is stored using the basic algorithm.

EXPERIMENTATIONS

In this section, we demonstrate the effectiveness of our algorithm through experimental analysis. The experiments measure the different aspects of the algorithm

as follows. We measure the capacity that a particular cover image can offer. Here, higher capacity means that lower fraction of the cover image is utilized by a data file, while maintaining high level of security, compared to other similar algorithms. Statistical measure of distortions is carried out through Signal to Noise ratio (PSNR) analysis. Effectiveness of the statistical estimations of the capacity of a particular cover image is measured. For this purpose, we estimate the capacity of different cover images and evaluate the estimations by measuring the actual number of bits required to store the data files.

Figure 3 shows the different cover images used in our experimentations. Figure 3 (a), (c) and (d) are bitmaps with resolution 640 X 480, while the resolution for Fig. 3(b) is 640 X 427, for Fig. 3(e) is 300 X 300 and for Fig. 3(f) is 497 X 480.

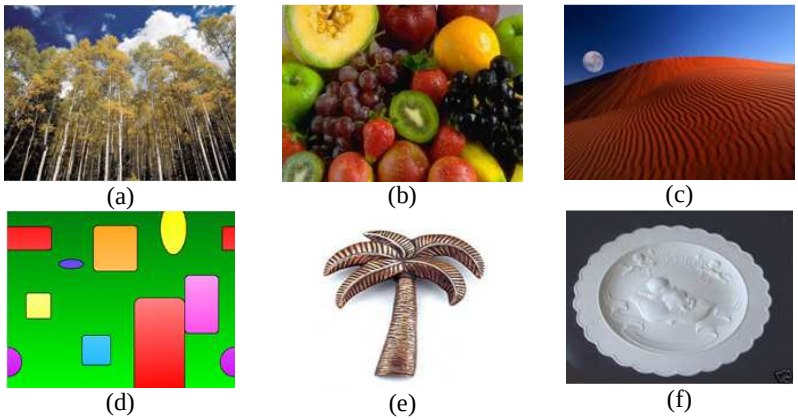


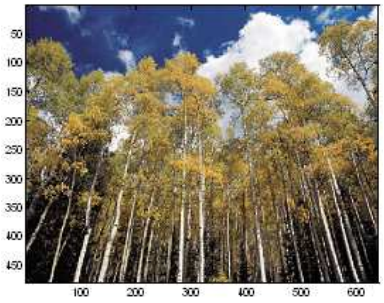
Figure 3: Bitmap images used in the experimentations: (a) forest, (b) fruits, (c) desert, (d) geometry, (e) palm-tree and (f) baby-dish.

The Basic Algorithm

First, we show the capacity utilization by the basic algorithm for a particular data file. Here, it is assumed that a particular partition scheme is already selected. The data file selected for this analysis is shown in Fig. 4.

Figure 4: Illustration of a data file: image size is 150 X 117, bit depth: 24.

Figure 5 shows the stego files for four different partition schemes. The value of the key is 17 (chosen arbitrarily for simplicity) and the indicator sequence is generated randomly using this shared key.

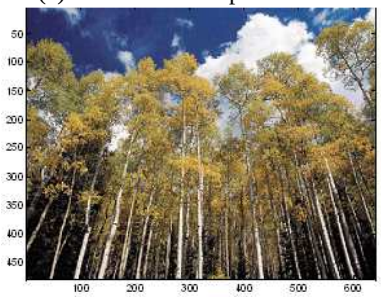


Pixels utilized in cover media: 50939
Partition Scheme: [256, 256, 0, 0, 0, 0, 0, 0]

(a) Constant 3 bits per channel

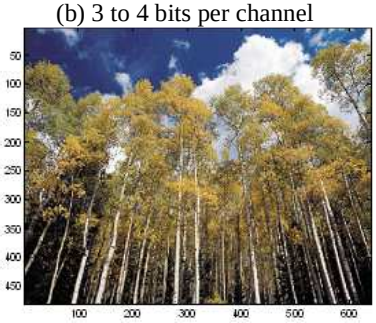
Pixels utilized in cover media: 41061
Partition Scheme: [256, 256, 96, 0, 0, 0, 0, 0]

(b) 3 to 4 bits per channel



Pixels utilized in cover media: 38364
Partition Scheme:[256, 256, 256, 0, 0, 0, 0, 0]

(c) Constant 4 bits per channel



Pixels utilized in cover media: 35791
Partition Scheme: [256, 256, 256, 48, 0, 0, 0, 0]

(d) 4 or 5 bits per channel

Figure 5: **Stego file for four different partition schemes using the basic algorithm.**

Table 1 shows the comparative results of our algorithm with the pixel indicator technique (Gutub et al. 2008).

Table 1: Comparison of performance of two steganography algorithms.

Technique	Number of data bits per channel (bits)	Number of pixels of cover media utilized (pixels)	Number of pixels of cover media utilized (percentage)
Our Basic Algorithm	3	50939	16.58%
	3 or 4	41061	13.37%
	4	38364	12.49%
	4 or 5	35791	11.65%
Pixel Indicator	2 + 2	77578	25.25%
	3 + 3	59051	19.22%
	4 + 4	44687	14.55%

The second column of Table 1 lists the number of bits of a particular channel modified to store the data bits. Here *m* + *m* (Table 1, second column, related to Pixel Indicator) means that two channels of a particular pixel are used for storing data and

each channel stores *m* data bits. The percentages shown in the last column is calculated as the ratio of the number of pixels of the cover image utilized to store the data and the actual number of pixels in the original cover image.

Table 1 clearly shows that, intensity based variable-bits algorithm performs better than the pixel indicator algorithm in terms of capacity utilizations. For comparison purpose, we have modified the pixel indicator algorithm to store up to 3 or 4 data bits per channel. But this modification leads to visual distortions to the cover image making it unpractical for other than comparison and study usage.

Our algorithm ensures a minimum capacity for any cover image, whereas for pixel indicator method may sometimes become very low. As an example, for the cover image of Fig. 3(d) and data file of Fig. 4, our algorithm uses only 16.5% of the capacity of the cover image (fixing 3 bits data per channel), whereas pixel indicator method is not able to store the data file due to capacity shortage!

Adaptive Partition Scheme

The idea of data spreading is incorporated in our algorithm by adaptively selecting a partition scheme. We allow a set of partition schemes to be used in a communication scenario and the 'best' partition scheme is selected adaptively to obtain the lowest possible distortion. Here, the *best* partition scheme is the one which allows the data file to utilize the capacity of the cover image as much as possible. To keep the algorithm simple, we assume that a set of partition schemes is defined beforehand in a particular communication scenario. The best scheme is selected based on the statistical capacity estimation formula as defined previously.

In our experimentations, we use the following set of partition schemes, from which the best partition is chosen at run time. The partitions are ordered in a sense that the partition with lower number will use the higher number of pixels of the cover image.

Table 2: Partition schemes used in the experimentations.

Partition Number	Partition								Bits/Channel
1	0	0	0	0	0	0	0	0	Constant 1
2	128	0	0	0	0	0	0	0	1 or 2
3	256	0	0	0	0	0	0	0	Constant 2
4	256	256	0	0	0	0	0	0	Constant 3
5	256	256	96	0	0	0	0	0	3 or 4
6	256	256	256	0	0	0	0	0	Constant 4

7	256	256	256	48	0	0	0	0	4 or 5
8	256	256	256	160	128	0	0	0	4, 5 or 6

Effectiveness of Statistical Estimations

When the secret message is too large to be stored in the cover image, the proposed algorithm can guess that situation without trying to encode the data first. The statistical estimation of capacity of the cover image can prove very handy in these cases. To demonstrate this case, a higher resolution version of Fig. 4 having 570,224 bits of data is used as the secret message. The image in Fig. 3(e) is used as the cover media. This secret message is selected so that that data file length is too big to store within the cover image, for the partition schemes of Table 2.

Table 3 shows the estimated maximum number of bits that can be stored for each partition scheme. These estimations are calculated using the formula described before. Table 3 shows that the capacity of the cover image in Fig. 3(e) is insufficient to store the selected secret message.

Table 3: Statistical capacity estimations by the proposed algorithm.

Partition Scheme	**Bits per channel**	**Estimated Max. Number of Bits that can be Stored**
0 0 0 0 0 0 0 0	1	90,000
128 0 0 0 0 0 0 0	1 or 2	135,000
256 0 0 0 0 0 0 0	2	180,000
256 256 0 0 0 0 0 0	3	270,000
256 256 96 0 0 0 0 0	3 or 4	303,750
256 256 256 0 0 0 0 0	4	360,000
256 256 256 48 0 0 0 0	4 or 5	376,875

Reducing Distortion through Data Spreading

Data spreading through adaptive selection of partition scheme brings two benefits. First, it allows the data to be spread throughout the cover image, which in turn reduces distortion. Second, dynamic (run time) selection of partition schemes relieves the burden of finding appropriate partition scheme for each cover image and data file pair. Only the shared key needs to be agreed upon by sender and receiver through a key distribution scheme like Diffie and Hellman (1976). The partition scheme is transmitted along with the data and there is no overhead for partition schemes management.

To test the imperceptibility performance of our algorithm, we generate randomly a message of length 1,50,000 bits with uniform distribution. We hide this message in

the cover images of Fig. 3 using the partition schemes of Table 2. To measure the distortion, we evaluate peak signal to noise ratio (PSNR) (Jain 1989) for all the stego images:

$$PSNR = 10 \times \log\left(\frac{255^2}{MSE}\right)(dB),$$

where, MSE (mean square error) is the error between cover image c (with m x n resolution) and stego image s:

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

Table 4 and 5 show the PNSR values and pixel utilizations for the different cover images and partition schemes. Using lower numbered partition means spreading the data more. It is clear from Table 4 that the proposed algorithm achieves significant improvement in reducing distortion by spreading the data. Higher PSNR is obtained when the secret message is spread over the media by selecting appropriate partition scheme. Moreover, for all the cover images and partition schemes, PSNR is more than 37 dB. This means that the proposed steganography algorithm provides very good imperceptibility performance.

Table 4: PSNR values (in dB) for different stego-images created by the proposed method.

Partition Scheme	PSNR (in dB)					
	Forest	Fruits	Desert	Geometry	Palm-tree	Baby-dish
8	42.38	46.64	44.21	65.29	40.36	39.19
7	52.18	50.76	49.48	55.38	42.33	51.14
6	50.94	52.11	52.54	57.71	42.49	48.04
5	51.80	54.03	53.14	58.68	47.03	48.25
4	55.72	56.89	56.67	61.43	47.61	54.16
3	59.83	60.78	60.28	64.64	51.16	58.70
2	60.54	61.90	60.52	65.15	51.16	59.03
1	62.52	62.55	62.70	67.11	51.16	60.58

Table 5: Utilization of pixels for different cover images by the proposed method.

Partition Scheme	Pixel Utilized (in %)					
	Forest	Fruits	Desert	Geometry	Palm-tree	Baby-dish
8	8.75	9.88	8.41	8.45	37.20	10.75
7	11.58	12.48	10.94	10.06	41.07	14.00
6	12.42	13.96	12.42	12.42	42.00	15.93
5	13.28	14.98	12.83	12.59	52.54	16.53
4	16.48	18.53	16.48	16.48	55.89	21.17
3	24.62	27.68	24.62	24.62	83.67	31.65
2	29.31	30.77	25.56	25.57	83.67	41.51
1	49.03	55.12	49.04	49.04	83.67	63.09

CONCLUSION

In this paper, we proposed a novel approach for color image steganography by adaptively selecting the number of secret data bits to be stored in a pixel of the cover image. Our algorithm achieves its goal in two stages. It adaptively selects a partition scheme of the color intensity values to utilize the capacity of the cover image as much as possible. Once a partition scheme is selected, the number of bits that a pixel can store is determined by the actual intensity values of the channels. This novel approach leaded to high capacity steganographic algorithm generating stego images with very low distortions. We demonstrated that our algorithm performs better than other similar algorithms with the same secret data and cover image. In addition, our algorithm spreads out the secret message as much as possible by selecting an appropriate partition scheme. This dynamic data distribution results in a more secure system making it a very attractive scheme for color image steganography.

ACKNOWLEDGEMENT

We thank the anonymous referees for their reviews that significantly improved the presentation of this paper. We would also like to thank King Fahd University of Petroleum and Minerals (KFUPM) for partially hosting this research. Thanks to both research centers: Center Of Excellence in Information Assurance (CoEIA), King Saud University, Riyadh, and Center of Research Excellence in Hajj and Omrah, Umm Al-Qura University (UQU), Makkah , for collaborative moral support toward the achievements in this work.

REFERENCES

Anderson, R. J., Petitcolas, F.A.P. 1998. On the limits of steganography. IEEE Journal of Selected Areas in Communications. Special Issue on Copyright & Privacy Protection. 16(4):474 – 481.

Bailey, K., Curran, K. 2006. An evaluation of image based steganography methods using visual inspection and automated detection techniques. Multimedia Tools and Applications. 30(1):55-88.

Chang, C.C., Chan, C.S., Fan, Y.H. 2006. Image hiding scheme with modulus function and dynamic programming strategy on partitioned pixels. Pattern Recognition 39 (6):1155–1167.

Chen, Y.F., Chien, S.W., Lin, H.H. 2009. True color image steganography using palette and minimum spanning tree. Proceedings of the 3rd WSEAS international conference on Computer engineering and applications.pp. 273-278.

Chen, W.Y. 2007. Color image steganography scheme using set partitioning in hierarchical trees coding, digital Fourier transform and adaptive phase modulation. Applied Mathematics and Computation. 185(1):432–448.

Chen, W.Y. 2008. Color image steganography scheme using DFT, SPIHT codec, and modified differential phase-shift keying techniques. Applied Mathematics and Computation. 196(1):40–54.

Cox, I., Miller, M., Bloom, J., Fridrich, J., Kalker, T. 2007. Digital Watermarking and Steganography. Second Ed. Morgan Kaufmann.

Diffie, W., Hellman, M.E. 1976. New directions in Cryptography. IEEE Trans. On Inform. Theory. 22:644 – 654.

EzStego. http://www.securityfocus.com/tools/586.

F5. http://wwwrn.inf.tu-dresden.de/%7Ewestfeld/f5.html.

Gutub, A., Ankeer, M., Abu-Ghalioun, M., Shaheen A., and Alvi, A. 2008. Pixel indicator high capacity technique for RGB image based steganography. Proceedings of WoSPA 2008 – 5th IEEE International Workshop on Signal Processing and its Applications.

J-Steg. http://www.ussrback.com/crypto/steanography/DOS/jsteg.txt.

Jain, A.K. 1989. Fundamentals of Digital Image Processing, Prentice-Hall, New Jersey.

Johnson, N. F., Jajodia, S. 1998. Exploring Steganography: seeing the unseen. IEEE Computer Magazine: 26-34.

Li, S.L., Leung, K.C., Cheng, L.M., Chan, C.K. 2006. A novel image-hiding scheme based on block difference. Pattern Recognition 39 (6):1168–1176.

Liu, C.L, Liao, S.R. 2008. High-performance JPEG steganography using complementary embedding strategy. Pattern Recognition. 41:2945 – 2955.

Noda, H., Niimi, M., Kawaguchi, E. 2006. High-performance JPEG steganography using quantization index modulation in DCT domain. Pattern Recognition Letters. 27 (5):455–461.

Parvez, M.T., Gutub, A.A. 2008. RGB intensity based variable-bits image steganography. APSCC 2008 – Proceedings of 3rd IEEE Asia-Pacific Services Computing Conference. pp. 1322 – 1327.

Provos, N., Honeyman, P. 2003. Hide and seek: an introduction to steganography. IEEE Security & Privacy Magazine. 1: 32–44.

S-Tools. ftp://ftp.ntua.gr/pub/crypt/mirrors/idea.sec.dsi.unimi.it/code/stools4.zip.

Sur, A., Goel, P., Mukhopadhyay, J. 2009. A novel steganographic algorithm resisting targeted steganalytic attacks on LSB matching. Digital Watermarking. Lecture Notes in Computer Science. Springer Berlin / Heidelberg. 5450:199–208.

Westfeld, A., Pfitzmann A. 2000. Attacks on steganographic systems. Proceedings of the Third International Workshop on Information Hiding. pp. 61–76.

Wu, H.C., Wang, H.C., Tsai, C.S., Wang, C.M. 2010. Reversible image steganographic scheme via predictive coding. Displays 31(1):35–43.

Yu, J.G., Yoon, E.J., Shin,S.H., Yoo, K.Y. 2008. A new image steganography based on 2k correction and edge-detection. Proceedings of the Fifth International Conference on Information Technology: New Generations. pp. 563–568.

Yu, Y.H., Chang, C.C., Hu, Y.C. 2005. Hiding secret data in images via predictive coding. Pattern Recognition. 38 (5):691–705.