ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

An Efficient Image Steganography Scheme using Tree Based Parity Check

Syama Krishna S[1], Prof. Ambikadevi Amma T[2], Shijoe Jose[3]

PG Student, Department of Computer Science and Engineering, JCET, Palakkad, Kerala, India[1]

HOD, Department of Computer Science and Engineering, JCET, Palakkad, Kerala, India[2]

Assistant Professor, Department of Computer Science and Engineering, JCET, Palakkad, Kerala, India[3]

ABSTRACT: Steganography is the practice of data hiding within some other information. The important fact is that the existence of the message is known only to the recipient. The data can be hided in different media such as audio, video, text etc. But the most popular and secured method of steganography is by using image as cover medium. In most of the methods data is concealed in randomly selected pixels of an image. An existing image is used as cover medium in most of the image steganographic methods which results in many drawbacks. Here image steganography is done using texture images. Texture plays a crucial role in many applications such as computer graphics, computer vision, image processing etc. The texture synthesis process aims at constructing a larger arbitrary sized texture image from a smaller sample of it. The resulting image has a local appearance as that of the sample input image. In order to increase the security a Tree Based Parity Check (TBPC) algorithm is used to hide data in the cover image.

KEYWORDS: Data embedding, steganography, TBPC, texture.

I. INTRODUCTION

Information security has been one of the challenging problems in data communication and information technology presently since the rise of the Internet. The terms cryptography and steganography is closely related. Cryptography is an important method of network security for securing the secret information transmitted through the communication channel. In cryptography, the secret message to be hided is encrypted using some method. It is then transmitted through the network channel and at the other end it is decrypted, finally obtaining the secret message. One of the drawbacks of cryptography is that there is no such way to hide the existence of the secret message. Using steganography this can be achieved. The term cover image is used for the image in which data is to be concealed. The main feature of steganography is that the attacker is unaware of the existence of secret message. As the size of the embedded information increases the strength of the steganographic approach reduces since the image gets more and more distorted.

An existing image is used as cover medium in most of the steganographic methods. One of the problems is that this can be easily cracked using various steganalysis techniques. This is because the image after embedding data, which is usually called a stegoimage will definitely contain some distortions. This will in turn reflect the artifacts of the image. Another problem in using an existing image as cover medium is that the image distortion increases as the amount of secret message to be embedded increases. This distortion can be reduced by considering a texture image as a cover medium instead of an existing image. It can be further reduced by structuring the texture image into a tree based form. In this paper, a steganography has been proposed which combines both texture synthesis as well as tree based parity check method.

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

II. RELATED WORK

Secret message can be hided in RGB 24 bit image. This can be done through various algorithms, one being the Optimized Bit Plane Splicing algorithm [1], [4], [10] and using data structures such as linked lists [1], [10]. Least significant bit (LSB) insertion [3] is a common, simple approach to embed data in different media. But it is highly risky. Masking and filtering techniques [3], is another method to hide data by marking an image. This is usually restricted to 24-bit and gray-scale images. Steganography can also be implemented in transform domain [2]. Several algorithms are used for this. JSteg algorithm sequentially replaces the least-significant bit of DCT coefficients with the message's data. OutGuess 0.1 is another transform technique that uses a pseudo-random number generator to select DCT coefficients at random. A new steganographic scheme for 3D point cloud models using self-similarity partition was introduced[11]. Reversible data hiding [12], in which the stego-media can be reversed to the original cover media exactly, has attracted increasing interests from the data hiding community. Texture synthesis is important for many applications in computer graphics, vision, and image processing. However, it remains difficult to design an algorithm that is both efficient and capable of generating high quality results. Texture synthesis is mainly of two types, pixel based and patch based. A simple image-based method of generating novel visual appearance in which a new image is synthesized by stitching together small patches of existing images was developed. This process is called imagequilting[5]. One of the advantages is that it is very fast and simple. Second, the algorithm is extended to perform texture transfer – rendering an object with a texture taken from a different object. A new patch-based texture synthesis method have been introduced [9],[13]. The core of the proposed method consists of two main components: (1) a feature-weighted similarity measurement to search for the best match and (2) a dynamically prioritized-based pixel re-synthesis to reduce discontinuity at the boundary of adjacent patches.

III. PROPOSED WORK

The proposed method is illustrated in this section. First, we will define some basic terminology to be used in our algorithm. A source texture is given as input. An image block of this source texture is considered as a patch. The size of the patch is user-specific. Kernel region is the central part of the patch and boundary region is the portion around the kernel region. $P_w$and P_h are the width and height of the patch respectively. $P_d$is the depth of the boundary region. Another concept used is the kernel block which is of size $K_w \times K_h$.

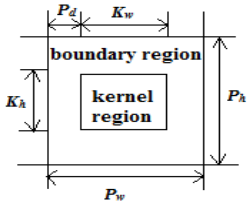

Fig. 1. Diagram of a patch

A source texture has dimensions $S_w \times S_h$. Kernel blocks are non-overlapped blocks obtained by dividing the source texture. Source patch is another important concept. It is obtained by expanding a kernel block upto a depth of P_d. When the kernel blocks are expanded this will result in the overlapping between them.SP_ndenotes the number of source patches. It canbe obtained using (1).

$$SP_n = S_w / K_w \times S_h / K_h \qquad (1)$$

where$K_w \times K_h$ is the size of kernel block and $S_w \times S_h$ is the size of the source texture.

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

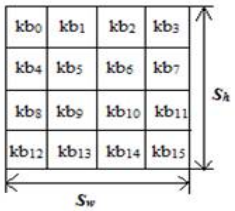

Fig. 2. An illustration of non-overlapped kernel blocks subdivided from the source texture.

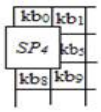

Fig. 3.The diagram of source patches derived by the expanding process using kernel blocks.

The input source texture undergoes the texture synthesis process and finally gets a new synthetic texture. While performing this, there is a need to generate candidate patches. Candidate patches are those into which we conceal the secret message. The source patches are only used to synthesise a new texture. A care must be taken to ensure that each candidate patch, CP, is unique. Otherwise, an incorrect message may be retrieved. In order to ensure the uniqueness, the flag procedure is used. The flag factor will be on for the firstly generated candidate patch and off for further duplicate candidate patches. Candidate patches are generated by considering the pixels of the source texture, travelling along the pixels in a scan-line order by employing a window of dimension $P_w \times P_h$.

A. Sender Side

The proposed method can be explained in four processes namely, creation of index table, creation of composition image, texture synthesis process, and concealment of secret message. The secret message is initially encrypted using the AES technique. This encrypted secret message is then subjected to Tree Based Parity Check (TBPC) method. Tree Based Parity Check method again consists of several steps namely, creation of primary, creation of secondary and creation of stego tree. In this manner, the stego code is obtained. The stego code is then merged with the synthetic texture. The size of the synthetic texture is directly proportional to the embedding capacity. The message embedding procedure at the sender side is shown below.

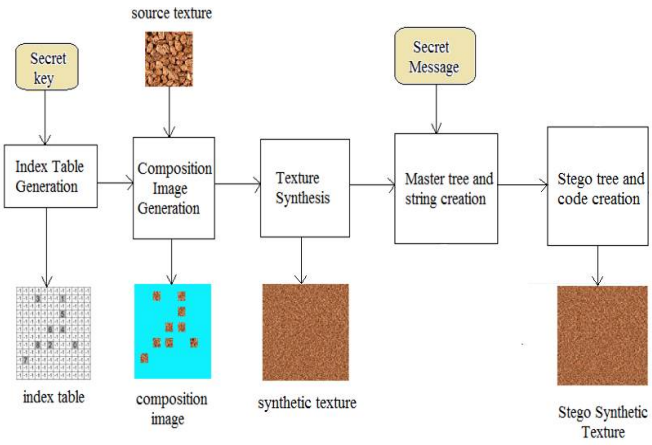

Fig. 4. Message embedding procedure

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

1) Creation of index table:

The source patch set SP obtained from the source texture is distributed in the synthetic texture. The positions at which these are to be distributed is given by index table entries which are in turn obtained by choosing a random seed. By referring to the index table, the initial source texture can be retrieved completely. Initial values for index table entries is indicated by -1. When the source patches are distributed in the synthetic texture, the value of -1 in the index table will be replaced by the respective source patch ID. The remaining entries are not changed. The value -1 shows the table is blank. The source patches can be distributed either in a sparse manner or in a dense manner. If the objective of our method is to conceal large amount of information, the source patches are distributed in a sparse manner. If the objective is high security to the data, then source patches are distributed in a dense manner. However, they are pasted in a random fashion, which ensures high security to our method.

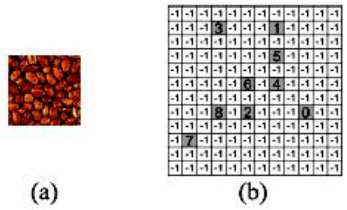

Fig. 5. Index table

For example, consider there are 16 source patches and synthetic texture has a total of 144 patches. The secret message will be concealed in 128 blank positions. One of the highlighting benefit of the method is its reversibility. The initial source texture can be retrieved at the receiver side by referring to the index table. The source patches are avoided from being distributed on the borders of synthetic texture. This will improve the image quality. If there are a total of 16 source patches, they are assigned IDs starting from 0 to 15.

2) Creation of composition image

In this process, a blank image is first taken which has the same dimensions as that of the synthetic texture. The source patches thus generated is pasted on to this blank image by looking into the source patch IDs in the index table. An utmost care should be taken while pasting the source patches since they may get overlapped.

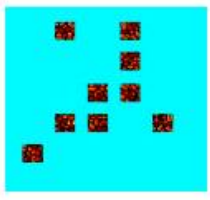

Fig. 6. Composition image

3) Texture Synthesis Process

Texture synthesis can be done in two ways either pixel based or patch based. The proposed method is developed using patch based texture synthesis. The method synthesis a new image by stitching together small patches from the sample image. The method in synthesis result an image block by block in raster order. Square blocks are used to capture the primary pattern in the sample texture. A block is randomly selected from the sample image and pasted into the new image beginning at the first row and the first column. Then another block is selected as a candidate neighbour. It is placed next to the first block so that they overlap one another.After the synthetic texture is produced, we go for the

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

selection of candidate patches upon which the secret message is to be concealed. The candidate patches are also selected in a random fashion. This selection is based on a secret key. The key can be generated using any of the encryption standards such as AES. The key is shared among the sender and the receiver.

4) Concealment of secret message

The secret message can be embedded via the Tree Based Parity Check (TBPC) method to produce the final stego synthetic texture. The TBPC method is a least significant bit steganography method. This method is used for generating stego code using the LSB of selected pixels which consist of following three steps.

Step 1:Construction of primary tree and creation of primary string and secondary string

A primary tree is created initially which is a complete binary tree. The length, *l*, of the secret message in binary form is calculated. The number of leaf nodes of the primary tree is equal to length *l* .The pixels of the selected candidate patch is taken and its Least Significant Bits are found out. The nodes in the primary tree is filled with these LSBs in the usual binary tree insertion manner. An even parity check is performed on the primary tree from root to leaf nodes thus generating the primary string. Secondary string is obtained by performing an Exclusive-OR(X-OR) between the message string and primary string. For example, consider the message to be concealed is 1011011(binary form). If the selected pixels for concealment are 154, 157, 152, 163, 150, 162, 152, 151, 160, 161, 165, 171, 168. The LSB of each pixel is used for creating the primary tree. The primary tree is shown below.

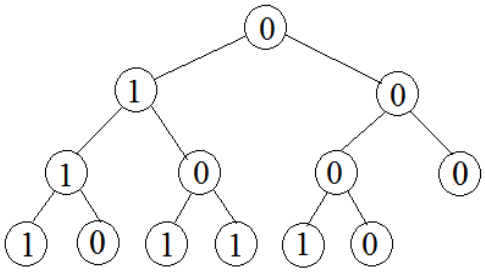

Fig. 7. Primary tree

After performing parity check, the primary string will be obtained as 1000100. After X-ORing between message string and primary string, the secondary string will be obtained as 0011111.

Step 2:Creation of secondary tree

Secondary tree is created by filling the leaf nodes with the secondary string and all the other non-leaf nodes with 0. Then this secondary tree is traversed in a bottom-up manner. If a particular non-leaf node has both its child nodes as one, they are flipped. Now the non-leaf node will be one and leaf node become zero. The secondary tree for the above example is shown below.

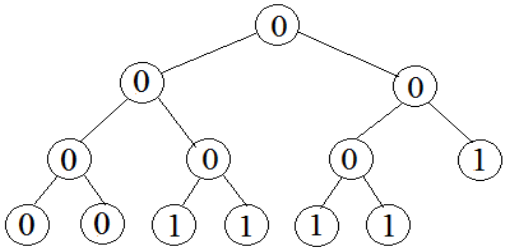

Fig. 8. Secondary string

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

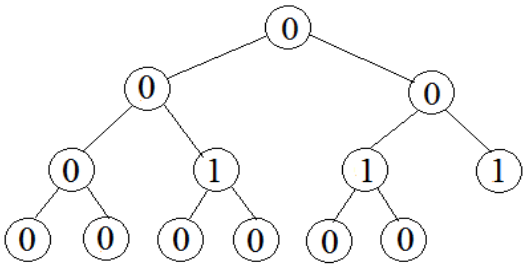

Fig. 9. Secondary tree

Step 3:Construction of stego tree

Now an X-OR is performed between the primary and secondary tree to finally obtained the stego tree. The stego tree for the above example is as follows.

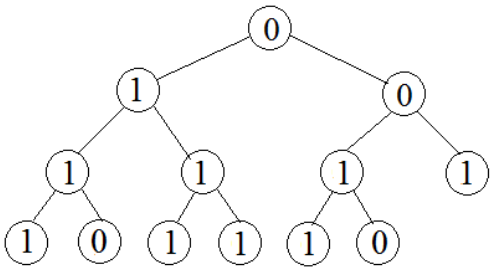

Fig. 10.Stego tree

The nodes of the stego tree represent the stego code. In the example, the stego code is 0101111101110. The stego code is then merged with the synthetic texture which was obtained after the texture synthesis. Thus the stego synthetic texture is obtained.

B. Receiver side

The receiver side includes four processes namely reverse index table generation, source texture recovery, composition image generation, and reverse Tree Based parity Check. The stego synthetic texture is the input at the receiver side. Given the secret key the same index table as the embedding procedure can be generated. The next step is the source texture recovery. Each kernel region with the size of $K_w \times K_h$ and its corresponding order with respect to the size of $S_w \times S_h$ source texture can be retrieved by referring to the index table with the dimensions $T_{pw} \times T_{ph}$. The advantage of source texture recovery is reusability. The same source texture can be again used for a second round of texture synthesis or steganography. In the third step, the composition image generation is applied to paste the source patches into a workbench to produce a composition image.

The final step involves reverse Tree Based Parity Check (TBPC). The reverse index table generation and composition image generation process is same as the sender side. The stego code is then extracted from the stego synthetic texture. This stego code is subjected to reverse TBPC. The reverse TBPC consists of three processes namely construction of stego tree, creation of secondary tree and creation of primary tree. The stego tree is constructed by filling a binary tree with the stego code. The primary tree is obtained in the same manner as the TBPC method. By performing X-OR between the stego tree and primary tree, the secondary tree can be obtained. Then this secondary tree is traversed in a bottom-up manner. If a particular non-leaf node has both its child nodes as zero, they are flipped. Now the non-leaf node will be zero and leaf nodes become one.These leaf nodes become the secondary string. After performing X-OR between the primary and secondary string, the message string is obtained. This is then decrypted using reverse AES to obtain the secret message. The message extraction procedure at the sender side is shown below.

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

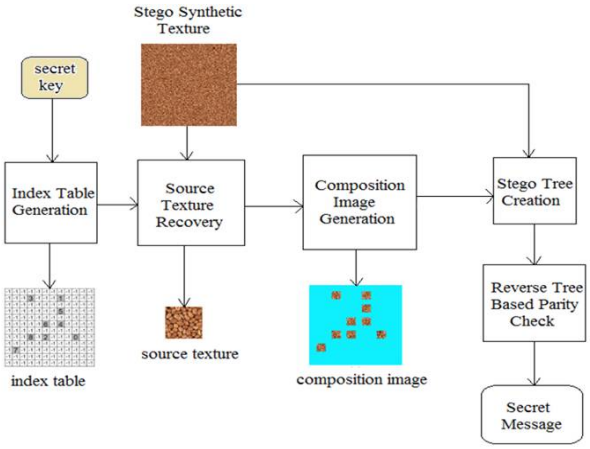

Fig. 11. Message extraction procedure

The reverse index table generation and composition image generation is same as the sender side. The stego code is then extracted from the stego texture. This is subjected to reverse TBPC. The reverse TBPC consists of three processes namely construction of stego tree, creation of secondary and primary tree. The stego tree is constructed by filling a binary tree with the stego code. By performing X-OR between the stego tree and primary tree, the secondary tree can be obtained. Then this secondary tree is traversed in a bottom-up manner. If a particular non-leaf node has both its child nodes as zero, they are flipped. Now the non-leaf node will be zero and leaf nodes become one.These leaf nodes become the secondary string. After performing X-OR between the primary and secondary string, the message string is obtained. This is then decrypted using reverse AES to obtain the secret message.

IV. EXPERIMENTAL RESULTS

Experiments were performed on three source textures namely, rope net, metal, and peanuts. Analysis was done on the basis of total embedding capacity and the computing time. For a fixed number of BPP,if the source texture has large resolution then the corresponding embedding capacity will be less and vice-versa.Image quality is also compared by MSE(Mean Square Error) method. This is shown in the following figures.

(a) Source texture  Rope net (192×192)  Metal (192×192)  Peanuts (96×96)

(b) Pure synthetic texture 

(c) 5 BPP 

Fig. 12 (a) Source texture (b) pure synthetic texture (c) stego texture (5 BPP)

ISSN(Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

Table I compares the embedding capacity to the MSE values. For the same patch size, the quality of the final image decreases as more secret messages are concealed in the image.

Table I
A comparison of embedding capacity to MSE

	Patch size = 24×24 Boundary depth = 4		Patch size = 48×48 Boundary depth = 8	
	Pure	5 BPP	Pure	5 BPP
Rope net	651.2	899.4	936.3	1224.2
Metal	595.6	795.2	837.4	1132.3
Peanuts	547.8	219.3	918.2	1034.6

V. CONCLUSION

This paper proposes a steganographic algorithm based on TBPC and texture synthesis. A large synthetic texture is produced from a small initial source texture. By using a conventional patch-based method the textures are synthesized. The method will also provides reversibility to retrieve the original source texture from the stego synthetic textures, making possible a second round of texture synthesis if needed. This paper also introduce another image steganography method that uses the Tree Based Parity Check algorithm to improve the image quality as well as to enhance the security of secret message. This also improves the embedding capacity to a large extent.

REFERENCES

[1] Komal Patel, SumitUtareja, Hitesh Gupta, " A Survey of Information Hiding Techniques", International Journal of Emerging Technology and Advanced Engineering,Vol 3, Issue 1, pp.347-350, January 2013
[2] N. Provos and P. Honeyman,"Hide and seek: An introduction to steganography," IEEE Security Privacy, vol. 1, no. 3, pp. 32–44, May/Jun. 2003.
[3] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," Computer, vol. 31, no. 2, pp. 26–34, 1998.
[4] M. Naseem, Ibrahim M. Hussain, M. Kamran Khan, Aisha Ajmal, "An Optimum Modified Bit Plane Splicing LSB Algorithm for Secret Data Hiding", Int. Journal of Computer Applications (0975 – 8887), vol. 29, no.12, pp.36-43, Sep 2011
[5] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn., pp. 341–346, 2001.
[6] Samir Kumar Bandyopadhyay, IndraKanta Maitra , "An Alternative Approach of Steganography using Reference Image", Int. Journal of Advancements in Technology,vol.1, no.1, pp.95-102, Jun. 2010
[7] Usha B A, Dr. N K Srinath, Dr. N K Cauvery, "Data Embedding Technique in Image Steganography Using Neural Network", Int. Journal of Advanced Research in Computer and Communication Engineering, vol. 2, no. 5, pp.2177-2180, May 2013
[8] Y.-M. Cheng and C.-M. Wang, "A high-capacity steganographic approach for 3D polygonal meshes," Vis. Comput., vol. 22, nos. 9–11, pp. 845–855, 2006.
[9] Chung-renYan,Tong-yee Lee, "Texture Synthesis with Prioritized Pixel Re-synthesis", Journal Of Information Science And Engineering , vol. 25, pp. 389-402, 2009
[10] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding-a survey," Proc. IEEE, vol. 87, no. 7, pp. 1062–1078, Jul. 1999
[11] Q Ke, Xie Dong-qing, "An High-capacity Steganographic Scheme for 3D Point Cloud Models Using Self-similarity Partition", International Journal of Modeling and Optimization, vol. 1, no. 1, pp.13-18, April 2011
[12] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354–362,Mar. 2006
[13] .M. F. Cohen, J. Shade, S. Hiller, and O. Deussen, "Wang tiles for Image and texture generation", ACM Trans. Graph, vol 22, no.3, pp. 287-294, 2003