# Enhancing Security in Steganography using Indexed Texture Synthesis and Color Synthesis

**Anusha P[1] Dr Y S Nijagunarya[2]**
[1]M.Tech Student [2]Professor
[1,2]Department of Computer Science Engineering
[1,2]Siddaganga Institute of Technology, Tumkur, Karnataka, India

*Abstract*— Digital Image Steganography is a techniques of hiding message or information in a digital image such a way that nobody can notice the existence of message. In this paper we using a novel approach for enhancing the security in steganography using Indexed texture synthesis. A texture synthesis process resample's smaller texture image to obtain a new image with same appearance and arbitrarily size. To conceal the secret message in image we use the texture synthesis in steganography. In contrast of using existing cover image to hide the secret message, our approach hides the source texture and embed the secret message through the process of indexed texture synthesis and color synthesis. This method has advantage over other steganography techniques, it offers the message embedding capacity is proportional to the size of stego image, it is difficult to reveal the secret message by an attacker since we are using dual indexed approach. The Experimental result shows theour approach provides various embedding capacities, provides visually plausible Stego image and extraction of source texture.

***Key words:*** Texture Synthesis; Color Synthesis; Steganography; Data Embedding

## I. INTRODUCTION

Many advances have been made in the region of steganography focusing on computerized pictures. Steganography[1] is a strategy of concealing message into the digital pictures or images.. The private covert communication between two parties using stego image should not arise doubt to eavesdropper that the image containing secret message [2], [4]. In recent years many steganographic algorithms have been proposed on digital images to enhance the security for stego synthetic images [5][6].

A large number of steganographic algorithms make use of cover image which leads to more distortion while embedding the secret message into the digital media. Since the size of the cover image is fixed it limits the size of the secret data to be embedded into the cover image. If the size of the secret message increases, it adds further distortion in the cover image and reducing the image quality. Hence, there is a tradeoff between size of the cover image and size of the embedding secret data. if the stego image contains more distortion the steganalyst can easily detect the presence of secret data present in the cover image. Steganalysis is a method to reveal the secret message hidden in digital media.

In this paper, we propose a new approach for enhancing security in steganography using Indexed texture synthesis and color synthesis. A texture synthesis is a method of reordering the smaller images to form new synthesized image such that image should appear same as of source image and arbitrary size and then using color synthesis the secret message is embedded on the color values of the synthesized image. In particular we avoid the use of cover image to embed the secret message.

The main benefits from our approach is, first we avoid the use of cover image .second ,we support arbitrary size of embedding secret data because we using texture synthesis .third, we using color index to embed the message in the RGB values of the image so It is difficult for the steganoanalyst to reveal the secret message. Experimental results show that our approach provides the embedding capacity equal to the size of the stego image. Theoretical analysis shows that it is difficult to find the secret data on image [6].

The paper is organized as follows: Section II describes the review of the texture synthesis with color index techniques. In Section III, detail description of the proposed algorithm including embedding and extracting procedures is presented. Section IV shows experimental results and theoretical analysis and is followed by conclusions and future.

## II. RELATED WORKS

Texture Synthesis and Analysis has an active research area in Computer Graphics and image processing .In recent years, many works has been carried out in texture synthesis to provide different texture synthesis algorithms and most of the algorithms uses Pixel/Patch based algorithm to generate the new image.

In 2000 Efros and Thomas [8][9] uses pixel based approach to generate the synthesized image pixel by pixel which uses the special neighborhood to choose the similar pixel in the sample image. But the problem with the pixel based approach is error propagation, if any one wrong selection of pixel will affect the consecutive pixels during synthesis time.

Otori and Kuriyama [10][11] works on pixel based texture synthesis to combine the both techniques such as texture synthesis and encoding to generate the stego image. In this approach they are going to embed the data which is encoded in the color values of the pixel pattern and placed on the blank image and then the rest of the image is filled by pixel synthesis. The overhead with this approach is error rate in extracting secret message from the stego image.

Cohen et al. and Xu et al made work on improving the image quality by using patch based texture synthesis approach[`12][13].In this approach the patches are pasted one another to form a new image and problem with this method is there exist a small overlapped during pasting procedure so overcome from this overhead later feathering approach is introduced by the Liang et al [6].

 

The latest work is done by Efros and Thomas [14][15]on Image Quilting for patch based synthesis. Image quilting is a process of stitching patches one another to form a new synthesized image and it is also called as Image Stitching approach.

In this techniques first selects the overlapped candidate patch in the image then we compute the error surface with newly chosen block and old overlapped block. Then by using dynamic programming technique we find the minimum cost at the overlapped surface of blocks. This repeats until we paste the blocks one another onto the texture.

Reversible image technique proposed by Ni et[16] which recover the original image after extracting the secret message from stego image. The current secret-of-the-art for reversible image data hiding is the general framework presented Li et al[16].In this paper, we present our approach which makes use of patch-based synthesizing procedure.

## III. PROPOSED WORK

Figure 1 shows the steps involved in texture steganography. First we read a secret key to generate the index table. The index table locates where the texture should be placed on stego synthetic image. The color index generation gives how to reorder the color values within texture image. The secret message is encrypted and is placed in the texture. The remaining empty space in the stego image is filled with the source texture to produce complete stego image.

The Basic terminology used in our Steganographic texture synthesis is called as "patch". patch is the image block in the source texture where its size is user-specific.the size of the patch is given by *Aw* (patch width) and *Ah*(patch height). The patch is divided into two regions such as Kernel region and boundary region .the central part is called as kernel region ,the region which is surrounded by kernel region is called boundary region with patch depth (*Ad*).

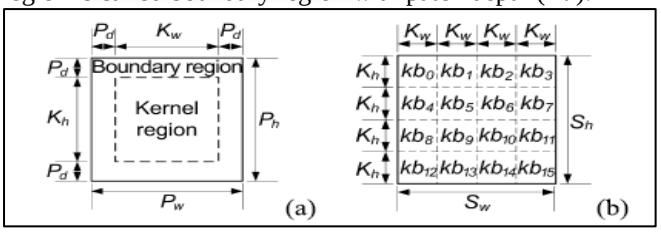


Fig. 1: (a) shows patch and Fig (b) shows source texture(*Sw * Sh*),it is divided into number of non-overlapped kernel blocks of size *Kw* and *Kh.*

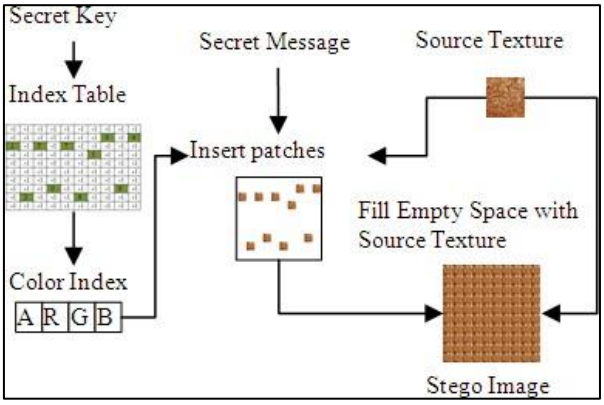


Fig. 2: The process of texture synthesis.

### A. Message Embedding Procedure

In this section, we briefly explain the message embedding procedure in a step by step approach.

### B. Index Table Generation

This Index table records the location information of source patch in the stego synthetic texture. To generate the Index table we need to determine the size of the Index table, the size of the table is given( *Iw**Patch size × *Ih**patch size ) where parameter *Iw* is texture width and *Ih* is Texture height. The number of entries in the Index table is calculated using the equation(1):

$$IAn = I\,Aw \times I\,Ah = [(Iw-Aw)/(Aw-Ad)+1] \times [(Ih-Ah)/(Ah-Ad)+1] \quad (1)$$

Once table is create ,First Index table is initialized with -1 this indicates that the table is empty then by calling the random number generator we are going to distribute the patch id into the Index table ,which gives the location information of source patch in the stego image. This is important step in embedding procedure because, since the Index table values is scattered and it difficult for the attacker to locate the patches to extract the secret message.

### C. Color Index Generation

In step 1 we scatter the patches, in step 2 we compute which color value holds which secret data. As we know the color information containing ARGB values (Alpha, Red, Green and Blue respectively), the alpha value referred as indexing value. According to the alpha value the secret data is embedded into RGB components. There are totally 6 combinations while reordering the RGB components. The value of the alpha component indicates the order in which the secret data is embedded in RGB component. For instance suppose if steganalyst is able to gather all the patches in order, but he/she may unable to obtain the original message back since the data is scattered in RGB components. Hence color indexing plays an important role in this approach.

### D. Patch Insertion Process

In first step we distribute the patch id, in patch insertion process first we create an empty image as our workbench and the size of the workbench is equal to the size of the stego image then we extract the patch from the source texture then by referring the index table position values, we paste the patches into the workbench. This process repeats for the patches The Figure 2 illustrates how patch insertion process works.

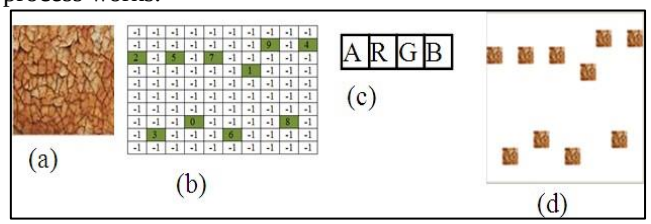


Fig. 2: Example of patch insertion process; (a) the source patch, (b) the index table for distributing patches, (c) the color index for distributing data, (d) the patch insertion into stego image.

### E. Message Embedding Process

A partial stego image which contains some scattered patches according to the index table is generated. Now, it's time to

 

embed data into scatter patches. First we select the patches in order with index table, second we read the color component, set the alpha component with color index, with reference to the color index scatter secret data value in RGB components. This process repeats for each data value from secret message.

After completing the process of embedding message, the empty part of the stego image is filled by patches of source texture to obtain complete stego image.

### *F. Message Extracting Procedure*

This is the reverse process of message embedding method. First, read the secret key to generate index table, second, extract all the patches from stego image with reference to the index table. Third, for each patch extracted, process each patch in scan line order to reach each pixel value, get the alpha value and use this value as index to read the RGB components in order to extract the secret byte. This process continues for all the patches extracted, combine the bytes to read the original secret message embedded within the stego image. The Figure 3 shows the step in message extraction process.

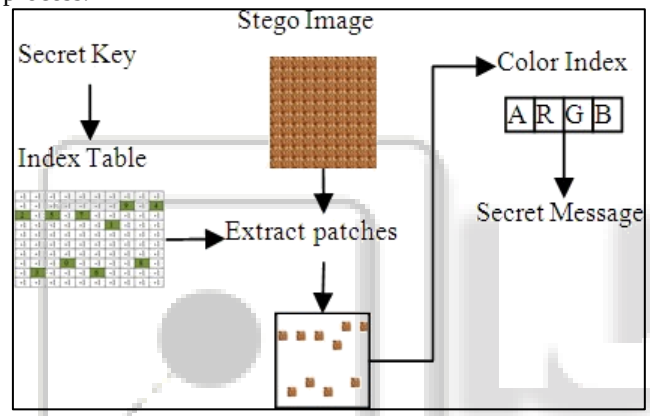


Fig. 3: Message Extraction Process.

## IV. EXPERIMENTAL RESULTS & ANALYSIS

The experiment is conducted with different texture images considering the size of the patch proportional to the size of the stego image, so that no overlapping of patches may occur. The maximum message embedding capacity provided by our approach is same as the size of the stego image. In Figure 4 the user selects the texture message. Figure 5 shows the secret message embedding with patch and color index. In Figure 6 we extract the secret message from stego Synthetic image.


Fig. 4: Read Texture Image

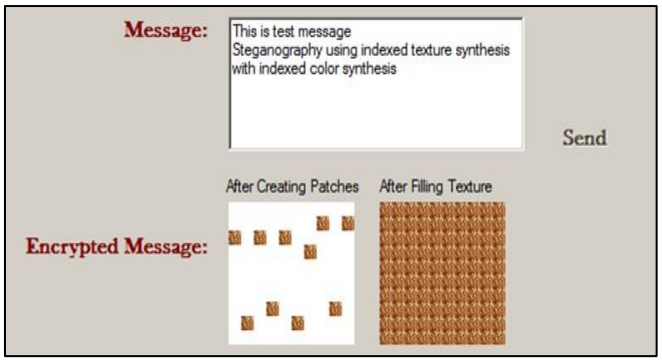


Fig. 5. Generate Stego image with Secret Data.

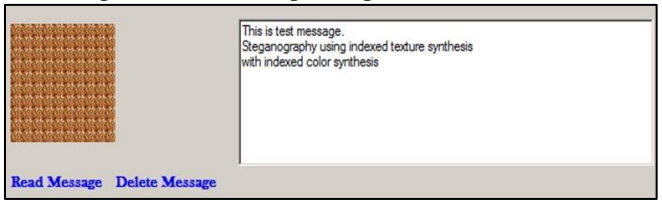


Fig. 6. Extract Secret Message.

### *A. Message Embedding Capacity Comparison*

Table I shows the comparison of message embedding capacities. We start with size of the stego image with 100X100, and patch size of 10X10. If we place one byte of message in each pixel of patch then patch can accommodate a total of 100 bytes. If one byte extended to two pixels it reduce the embedding capacity by half but enhance the security. After changing the patch size of 20X20 it does not going to effect the total embedding capacity.

| Texture Size: 100X100 Patch Size: 10X10<br>Total Number of Patches: 100 | | |
|---|---|---|
| Bytes per pixel | Total bytes per patch | Total bytes per Image |
| 1 byte per pixel | 100 | 10000 |
| 1 byte per 2 pixel | 50 | 5000 |
| 1 byte per 4 pixel | 25 | 2500 |
| Texture Size: 100X100 Patch Size: 20X20<br>Total Number of Patches: 25 | | |
| Bytes per pixel | Total bytes per patch | Total bytes per Image |
| 1 byte per pixel | 400 | 10000 |
| 1 byte per 2 pixel | 200 | 5000 |
| 1 byte per 4 pixel | 100 | 2500 |

Table 1: Total Message Embedding Capacity

## V. CONCLUSION & FUTURE WORK

We proposed steganography using indexed texture synthesis and indexed color synthesis to produce a stego image with secret embedded message. Till now no attempt has been made in combining both texture indexing and color indexing in patch steganography. This proposed method enhances the security for embedded secret message by maintaining a visually same stego image. It is hard for steganalyst to reveal the secret message embedded within stego image.

This work can be extended with other approaches to improve the quality of the image and also the embedding capacity.

## REFERENCES

[1] Kuo-Chen Wu and Chung-Ming Wang, "Steganography Using Reversible Texture Synthesis" *IEEE* Transactions On Image Processing, vol. 24, no. 1, pp. 131--139, 2015
[2] N. F. Johnson and S. Jajodia, ''Exploring steganography: Seeing the unseen,'' *Computer*, vol. 31, no. 2, pp. 26--34, 1998.
[3] N. Provos and P. Honeyman, ''Hide and seek: An introduction to steganography,'' *IEEE Security Privacy*, vol. 1, no. 3, pp. 32--44, 2003.
[4] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, ''Information hiding survey,'' *Proc. IEEE*, vol. 87, no. 7, pp. 1062--1078, Jul. 1999
[5] 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
[6] J. Fridrich, M. Goljan, and R. Du, ''Detecting LSB steganography in color, and gray-scale images,'' *IEEE MultiMedia*, vol. 8, no. 4, pp. 22---28, 2001.
[7] L.-Y. Wei and M. Levoy, ''Fast texture synthesis using tree-structured vector quantization,'' in *Proc. 27th Annu. Conf. Comput. Graph. Interact. Techn.*, 2000, pp. 479---488.
[8] A. A. Efros and T. K. Leung, ''Texture synthesis by non-parametric sampling,'' in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, Sep. 1999, pp. 1033---1038.
[9] C. Han, E. Risser, R. Ramamoorthi, and E. Grinspun, ''Multiscale texture synthesis,'' *ACM Trans. Graph.*, vol. 27, no. 3, 2008, Art. ID 51.
[10] C. Han, E. Risser, R. Ramamoorthi, and E. Grinspun, ''Multiscale texturesynthesis,'' *ACM Trans. Graph.*, vol. 27, no. 3, 2008, Art. ID 51.
[11] H. Otori and S. Kuriyama, ''Texture synthesis for mobile data communications,'' *IEEE Comput. Graph. Appl.*, vol. 29, no. 6, pp. 74---81,Nov./Dec. 2009.
[12] 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.
[13] K. Xu *et al.*, ''Feature-aligned shape texturing,'' *ACM Trans. Graph.*, vol.28, no. 5, 2009, Art. ID 108.
[14] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum, ''Real-time texture synthesis by patch-based sampling,'' *ACM Trans. Graph.*, vol. 20, no. 3, pp. 127---150, 2001.
[15] A. A. Efros and W. T. Freeman, ''Image quilting for texture synthesis and transfer,'' in *Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn.*, 2001, pp. 341---346.
[16] 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.

 