12th International Conference on Recent Development in Engineering Science, Humanities and Management

The International Centre Goa, Panjim, Goa (India)

14th April 2019 www.conferenceworld.in ISBN: 978-93-87793-85-9

Implementing indexed texture synthesis and indexed color synthesis in steganography

[1]Lokesh. S, [2]Aasim Kutub , [3]Arun Kumar , [4]Nivedita Pandey, [5]Raj Khare

[1]Assistant Professor ,Sai Vidya Institute of Technology, [2,3,4,5]CSE 8th semester

ABSTRACT

In this paper an approach for steganography using indexed texture synthesis and indexed color synthesis is used. A texture is generally a small image which is stacked accordingly to form a large image of same appearance. The secret message will be embedded in one of the textures. The advantage of using double indexed method to hide the secret message will be that it allows to embed the data which is proportional to the size of the image, it is very highly difficult to reveal the secret data by potential hackers. This method is different from other steganographic methods as general methods use cover image whereas this method uses indexed texture synthesis and indexed color synthesis.

Keywords: - Texture synthesis, color synthesis, steganography, data embedding.

I. INTRODUCTION

Steganography is a practice of hiding a secret message into a digital media in this case a digital image. The main purpose of this technique is to make private communication between two parties and no third party can eavesdrop between the two parties using the stego image. A stego image is an image where the secret message is embedded.

As of today there are many steganographic algorithms and while embedding the secret message into the image most of these algorithms make use of the cover image and the greatest disadvantage of using the cover is that, they lead to distortion in the stego image and can be easily detected that a secret message is embedded in it. Another disadvantage of cover image is that its size is fixed that means it can not vary or it can not increase. Since the size is fixed this puts limitation on the size of secret image that will be embedded. It at all the size of the cover image increases this reduces the image quality because as size increases it creates distortion. The steganalyst can easily analyze the stego image and can get the secret image out of it if the distortions produced by the cover image are high.

We have come up with a way to hide the secret message in the digital image using double indexed synthesis , that is indexed texture synthesis and indexed color synthesis. Texture synthesis basically is a method where small texture images are arranged in order and combined for similar local appearance and color synthesis is used for embedding the secret data using indexed color values.

The major advantages of double indexed synthesis is that:

1. We avoid the use of cover images.
2. Support of arbitrary size of secret data.
3. Use of color index in texture, which make very difficult for the steganalyst to retrieve the embedded secret data.

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 work.

II. RELATED WORKS

General steganographic algorithms mainly use pixel/patch to produce a stego image[1]. The idea behind pixel based approach is that a stego image is formed by synthesizing pixel by pixel using spatial relation to choose the next similar pixel in stego image. Suppose if at all any incorrect pixel is chosen while processing , the stego image may lead to error[2].

The pixel based approach coined by Otori and Kuriyama[3] suggests that the secret data is encoded in color valued pixel patterns and are placed on a blank image and the remaining is filled by pixel synthesis[4]. Though this approach is a good one but it led to more error rate in extracting secret messages from stego image.

The quality of the stego image was improved using patch based approach by Cohen et al and Xu et al[5], but this also came with a major disadvantage that is the patches caused overlapping of region within the stego image.

The solution for overlapping of region by patch based approach is given by Liang et al.[6] and the technique is called feathering technique. Two people named Efros and Freeman used technique known as image quilting[7] for patch synthesis. It picks the patch which is overlapped and also the calculates the error according to the neighboring patches then dynamically fix the boundary between the patches and synthesize the patch to obtain a stego image.

Ni et al[8] proposed the reversible image technique , this technique recovers the original image after the extracting the secret image from the stego image. Ni et al introduced the general framework for reversible image data hiding. We are making use of patch based synthesizing procedure.

III. PROPOSED METHOD

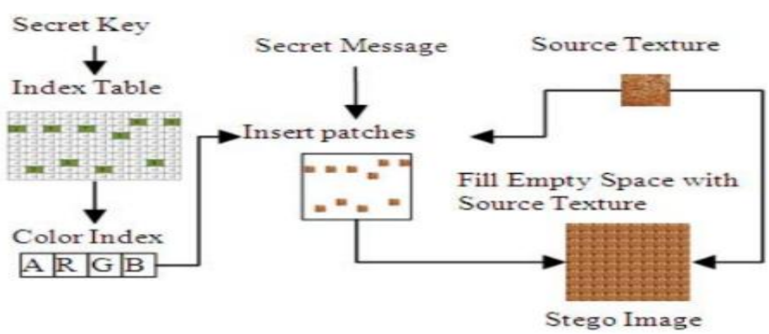

Fig: 1 Texture Synthesis process

The above figure clarifies the various process of texture synthesis, First of all the secret key is read to generate an index table for this key. This index table will be used to locate the texture in order to be placed on stego image. The function of color index is to arrange the color values within the texture image in proper order. After reordering the secret message is encrypted and is placed in the texture. Finally the stego image is completed by filling the empty spaces with the original texture.

The basic unit in this system is texture (patch). Patch is basically a small texture of the source image and the size of the patch is defined by the user and is specified by Aw (width) and Ah (height).

A. Message Insertion

This section explains the step by step procedure for message insertion.

1) Generating Index Table

In order to generate the index table, the location of each and every original patch which will be placed on the stego image should be noted/ recorded. This can be achieved while extracting the secret message from the texture. To do this firstly we have to find the size of index table and this can be using this ((Iw * PatchSize)*(Ih)/PatchSize)). Iw and Ih will be the width and height of the texture. To present the index table initially to be empty we have to initialize the index table to -1 then randomly generate the position values and add it to the table also which is the location of the patch in the stego image. Randomly generating the position values and adding in the table makes difficult for the hackers to locate the patches in sequence in order to extract the secret message.

2) Generating Color Index

In the previous step we placed randomly the patches and in the generation of color index step we will be computing which color value holds the secret data. The color information mainly contains the ARGB values this are nothing but Alpha, Red, Green, Blue respectively. The highest priority is given to the Alpha value and this value is also referred as indexing value. Based on this indexing value the data is embedded into the RGB components. Reordering the RGB components can produce specific 6 combinations. Alpha value specifies the correct order where the secret data will be embedded in the RGB component. Suppose a steganalyst gets hold of the order of all the patches, in that the person will not be able to obtain the original message back because the data is randomly scattered in the RGB components. That is why color indexing plays a major role.

3) Insertion of Patch

The main purpose now at this point is to put the patches into stego image. The number priority now is to create a plain empty image which will have the size equal to the stego image. Then the very next step will be to consider the index table and choose the location and place the patch in the empty image and the whole patch is inserted into the empty image. This process will be repeated until all the patches are inserted.

4) The Process of Message Insertion

Until now there was just an empty image but after the previous patch insertion process a partial stego image with scattered patches according to the index table is generated. Message insertion process will embed the message into the scatter patches. Each patch is selected in proper order by referring the index table, then the color component is read, right after that the alpha component is set according to the color index and with reference to this the secret data is scattered in RGB components. The whole process will be repeated for each data value from the secret message.

After embedding the message, the empty part of the stego image will be filled by patches of the source texture in order to generate the complete stego image.

B. Message Extraction

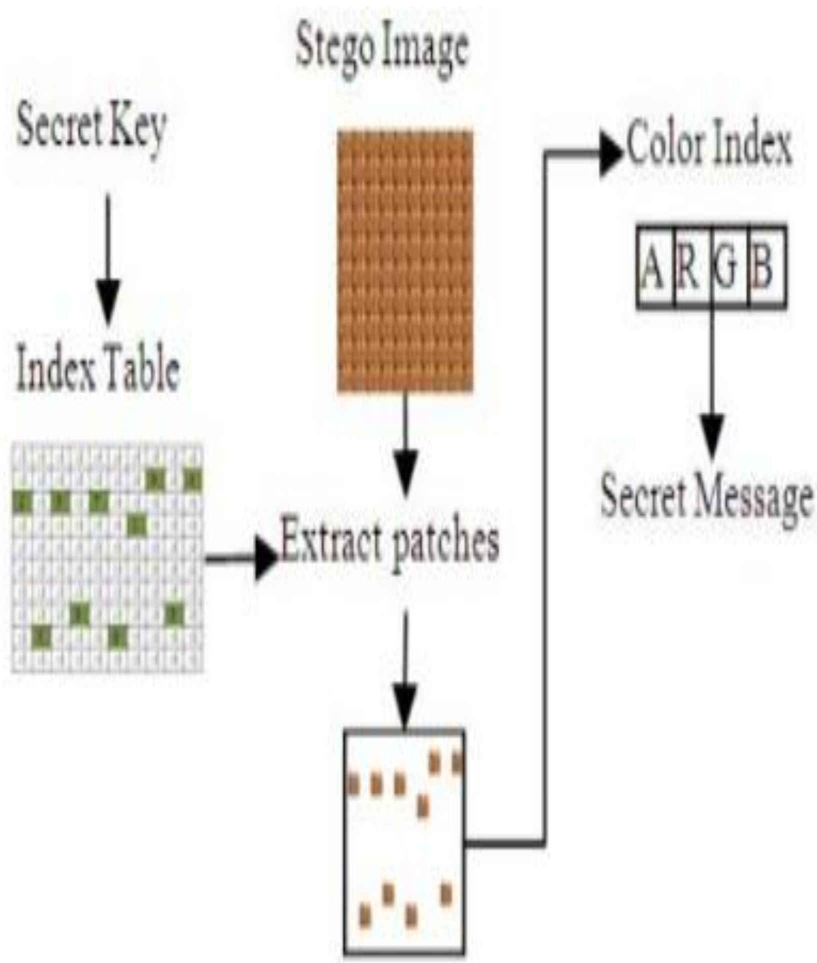

Fig: 2 Message Extraction Process

This is complete opposite process of message insertion. First thing for extracting the message from the stego image is to read the secret key and if the key is correct then the index table will be generated. Right after that all the patches has to be extracted by using the index table. Third step in this process will be to get the alpha value and this can be done by processing each patch that has been extracted in the previous step in a scan line order to reach at a point where we will get the pixel value, the get the alpha value and use this alpha value as index to read the RGB components in an order to secret byte. This process will be continued for all the patches extracted, then combine the bytes finally to read the original secret message embedded within the stego image.

IV. Conclusion

We proposed a method to hide vital information using indexed texture synthesis and indexed color synthesis to produce a new image which can be called as stego image. This proposed method thus enhances the security and will be very difficult for steganalyst to extract the secret message.

ISBN: 978-93-87793-85-9

REFERENCES

[1] Y. Guo, G. Zhao, Z. Zhou, and M. Pietikäinen, "Video texture synthesiswith multi-frame LBP-TOP and diffeomorphic growth model," IEEE Trans. Image Process., vol. 22, no. 10, pp. 3879–3891, Oct. 2013.

[2] L.-Y. Wei and M. Levoy, "Fast texture synthesis using tree-structuredvector quantization," in Proc. 27th Annu. Conf.Comput. Graph.Interact. Techn., 2000, pp. 479–488.

[3] H. Otori and S. Kuriyama, "Data-embeddable texture synthesis," inProc. 8th Int. Symp. Smart Graph., Kyoto, Japan, 2007, pp. 146–157.

[4] 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.

[5] K. Xu et al., "Feature-aligned shape texturing," ACM Trans. Graph., vol.28, no. 5, 2009, Art. ID 108.

[6] 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.

[7] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis andtransfer," in Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn.,2001, pp. 341–346.

[8] 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.