International Journal of Engineering Research and General Science Volume 4, Issue 2, March- April, 2016
ISSN 2091-2730

REVIEW ON STEGANOGRAPHY USING TEXTURE SYNTHESIS

Varnya C[1], Rinchu Baburaj[1], Harikrishnan N[2]
[1]PG Scholar, [2]Assistant Professor
Dept of ECE, Thejus Engineering College
Email:[1]1992ammuv5@gmail.com

Abstract— Steganography is the art of hiding data in a media such as image, video or audio files. This paper describes different texture synthesis mechanisms as well as steganographic techniques for the secure transmission of information with more capacity. Mainly, steganography is a type of an invisible communication which is recently being used in different applications. The data embedding capacity and the increased PSNR values are the major issues facing in steganography. To solve these issues, many algorithms have been put forward in these years. Texture synthesis is one of the efficient methods which can be used to improve the data embedding capacity. In this method, the data is hidden in the cover image which is generated by the method of texture synthesis which adds more data embedding capacity. It is a challenging task to generate high quality synthesis results. So different texture synthesis methods and data hiding schemes are analysed here.

Keywords— Cover Image, Data Embedding, Patch based synthesis, Pixel, Steganography, Stego Image, Texture synthesis, Tiling

1. INTRODUCTION

Steganography is a useful tool which allows covert transmission of information through the communication channel. comparing to cryptography, in steganography the data hidden will not be much visible to attackers. various formats of media such as audio, image and video can be used as cover for hiding the data[1]. But due to the ease of use and memory requirements, digital images are more popularly used as cover images for steganographic techniques.

Normally all digital file formats can be used for steganography as cover objects but an important property of redundancy is needed for hiding data. Redundancy can be explained as the amount of bits of an object that provide accuracy far greater than needed for the object's use and display. Redundant bits are those bit patterns in an image those does not disturb the visible features even if the bits are altered. Digital images[2] and audio files will obey this property but studies have found that other file formats can also support this redundancy. All digital file formats contain sequence of binary digits 0 and 1. So, the steganographic technique can be simply applied to the binary sequence by altering the one or two bits in the original sequence. The pixels of an image constitute the light intensities at specific points on the image. In an 8 bit image, each pixel will be an 8 bit sequence of 0's and 1's. So even if the least significant bit of the sequence is changed, it will not affect the entire image.

Texture is something which is composed of repeated patterns and it exactly look like a uniform image. Texture synthesis is the process of creating repeated patterns of textures from a smaller texture image known as the source texture by taking the advantage of its structural content. Texture synthesis finds its application in a wide variety of areas like graphics, image enhancement techniques, animation etc[3]. Nowadays steganographic data hiding schemes are also being used together with texture synthesis methods. By the addition of coding techniques and cryptographic techniques along with the above mentioned methods, the data hiding capacity and security[4] can be improved.

The entire paper is organized as follows: In section 2, different texture synthesis techniques are reviewed and finally out of which the most efficient technique is identified. Section 3 analyses various steganographic techniques in digital images and at last, the most useful data hiding technique is found. Section 4 describes methods that combine both steganography and texture synthesis and finally the better method is identified.

2. TEXTURE SYNTHESIS

Textures are images that contain some repeating patterns and also they will possess the possibility of certain random variations. The two important properties of a texture are:

- The generated texture output should look like input
- The algorithm that is used to generate the output must be easy to use

The generated texture synthesized image posses two properties such as local and stationary because each pixels in the texture image is visually related to a small set of its neighborhood pixels. Out of various algorithms proposed and presented, some of the texture synthesis methods are analyzed below.

2.1.Tiling Methods

This is one of the most simple and easy method of synthesizing a large texture from a smaller input texture. In this technique, multiple copies of the input sample are simply copied and pasted side by side to generate the output texture. The method presented by Cohen *et al.* [5] is a simple and stochastic tiling system for non-periodically tiling a plane using a number of Wang Tiles. The tiles then filled with textures, uniform images or uniform patterns will obtain a continuous representation. Wang Tiles are basically squares in which each edge is assigned a particular color which can be Red, Blue, Green and Yellow as shown in figure 2.1. Matching colored edges are aligned to tile the plane as shown in figure2.2. Tiling requires all neighboring edges between tiles to have matching colors. This

ISSN 2091-2730

method of tiling is named after Hao Wang in 1961 that any set of tiles that can produce a valid tiling of the plane must also be able to produce a periodic tiling of the plane.

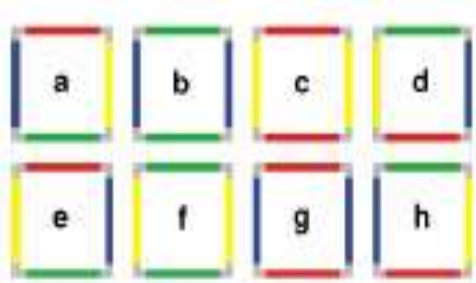

Fig 2.1: Eight Wang Tiles that can stochastically tile the plane [5]

New methods to fill the tiles with 2D texture, 2D Poisson distributions, or 3D geometry to efficiently create non-periodic texture as needed was presented. They also demonstrated how to fill individual tiles when combined with Poisson distributions that maintain their statistical properties.

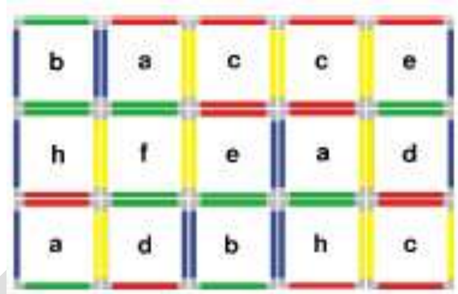

Fig 2.2: Align tiles to match edge color to create non-periodic tilings[5]

Edge coloring and the synthesis of textures ensures that the tiles always fit together. But a problem arises if an object or a visible artifact of the texture is placed across the tile corners. In this case, the vertical edge color constraint imposes all tiles with same colored opposite vertical edges to include the remainder of this object to ensure the fitting condition. Since the same object is also on the horizontal edge, all tiles that contain the corresponding horizontal edges needed to be placed similarly. This results in visible repetitive patterns which creates visual artifacts. To avoid this problem, a solution is found by coding the corners of a tile as an additional bit of information, by essentially coloring the corners as well as the edges. For example, in a single tile, $2^4 = 16$ possibilities of corner codings can occur. From their studies, they understood that the larger set of tiles provides increased degrees of freedom.

The method introduced by Weiming Dong *et al.*[6] is an optimized approach that can stably synthesize an ω-tile set of large pattern diversity and high quality. For that, an extendable rule is introduced to increase the number of sample patches to vary the patterns in a ω-tile set. In contrast to the other concurrent techniques that randomly choose sample patches for tile construction, the current technique uses Genetic Algorithm to select the feasible patches from the input example. This operation ensures the quality of the whole tile set. This is an approach for tile-based texture synthesis that is based on the optimization of tile set quality within a Genetic Algorithm (GA)-based framework.

2.2 Pixel – Based Methods

Due to the disadvantage aroused from tiling method because of the overlapping error and visible seams in the pasting edges, another method is introduced which is known as the pixel based texture synthesis. The method synthesizes a texture in scan line order by finding and copying pixels with the most similar local neighborhood.

A simple and efficient pixel based method is presented by Efros and Leung[7] in 1999. Their algorithm uses an input source image and a uniform white noise image which is having the size and dimension of the output texture. In this, a single pixel is generated at a time from an initial seed. A fixed size window with user specified size is taken which is centered on the currently synthesizing pixel. The more matching pixel is searched and copied in the output target. This process is repeated until all pixels in the target image are generated. This is illustrated in figure 2.3.

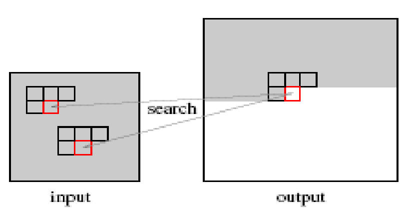

Fig 2.3: Pixel based synthesis

The paper presented by Wei and Levoy[8] uses a deterministic algorithm and the output texture is generated in a scan line order. This method improves the speed of the synthesis procedure by using tree structured vector quantization (TSVQ)[9] to match the target pixel with the neighborhood pixels. This fastest algorithm has a largest memory requirement. But it failed to synthesize the texture images of flowers, leafs, pebbles etc.

A slightly modified technique is introduced by Heeger and Bergen[10] which uses a multi resolution approach. The method uses an input texture and a modified noise image whose histogram is modified which matches with the input texture. The algorithm iterates by matching the noise histogram with the input texture image by using an image pyramid representation. It can be an image pyramid[11] or Laplacian pyramid. After some threshold iterations, the process can be stopped.

2.3 Patch based methods

In patch based methods, a single patch is generated at a time as shown in figure 2.4. In the previous method, only a single pixel is getting generated. So comparing to the existing methods, patch based method is much faster and is being widely used for texture synthesis applications[12].

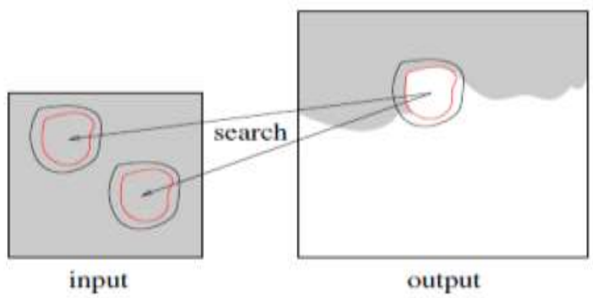

Fig 2.4: Patch based synthesis

Here the patches are copied block by block in a raster order. Upper left corner is initialized first. Then a square block is selected, which is taken as the candidate neighbour. Then next best matching patch is selected and is placed in the target image which overlaps with the other. Image quilting algorithm put forward by Efros and Freeman[13] minimizes the overlapping error introduced during the patch based texture synthesis. This implement a minimum error boundary cut in between the tiles copied as shown in figure 2. 5. The image quilting algorithm is as follows:

- Select a random patch from the input texture and initialize it as the seed to perform the texture synthesis and place it in the upper left corner.
- The process is done from left to right and top to bottom as:
  - Select the best fit patch to be pasted next using an exhaustive search.
  - Calculate the overlapping area between the current patch and the already processed patches using the L2 norm.

The minimum error boundary path is identified and add that new patch to the output texture

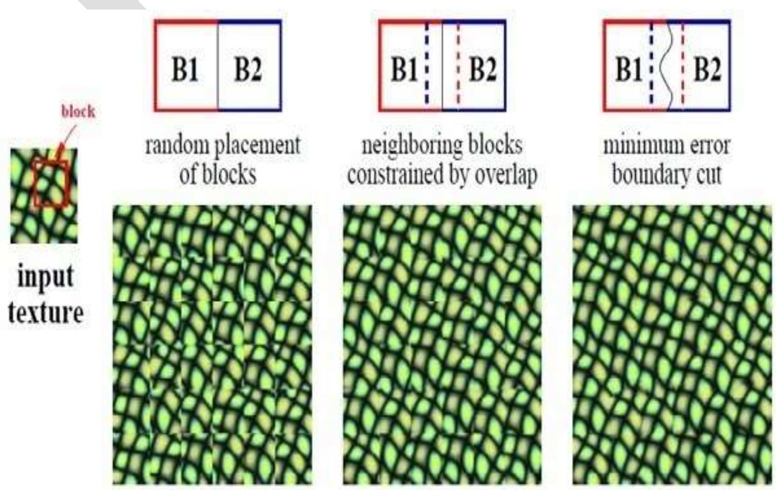

Fig 2.5: Quilting texture[13]

Liang, Liu and *et al*.[14] introduced a fast and new algorithm to generate texture using the method of texture synthesis. The texture generation was based on a non-parametric estimation of Markov Random Field [15] density function. This method assumes the texture as Markov Random model and the stochastic process is assumed to be stationary and local. Markov Random Field model is chosen because of its accuracy in modeling a variety of textures[16]. The method is both applicable to texture synthesis which depends on some constraints as well as which is independent of constraints. Constraint based texture synthesis is found in applications like texture synthesis for hole-filling and tileable texture synthesis. Here the constraint is a randomness parameter. The parameter is used to control the randomness of the synthetic texture. In this paper, an input sample texture 'Iin' is used to generate an output synthetic texture 'Iout' by making use of this patch based algorithm. A patch 'Bk' is selected from the input texture 'Iin' and is copied in the target image in each step to generate the output texture. For the ease of work, the patch size taken for pasting is taken a fixed size of 'WbxWb'.

By analyzing various steganographic techniques, patch based texture synthesis is found to be the most useful algorithm for synthesizing textures. This method generates a single patch at a time and thus it is a faster method. Also in order to avoid the visual artifacts and overlapping error, an image quilting algorithm is also used with the patch based texture synthesis method. Comparing to all other methods this technique is widely used.

3. STEGANOGRAPHY

Steganography is the technique of hiding secret data and can be called as a type of invisible communication. Steganography differs from cryptography in the fact that in the case of steganography, the source and stego image can't be easily distinguished. The technique which is used to hide the data must be selected such that even after hiding the data the media should not affect any kind of distortions. In order to hide data in images, the noisy or redundant areas[17] are to be found first. Because in that areas, while the bits of pixels are changed, it will not get noticed.

In the paper proposed by Johnson and Jajodia[18], steganographic techniques are explained in detail. Number of ways exist to hide the data in images. The most popular approaches are:

- LSB replacement
- Masking and filtering
- Algorithms and transformations

For a 24 bit image, 3 bits of data can be hidden in each byte of the pixels. If the data before hiding is compressed using coding and compression schemes, more amounts of data can be hidden in the cover image. The data is hidden by replacing the least significant bits by the bits in the secret data. The change in the least significant bit will not affect the cover image since LSB bits are redundant bits. So the resulting stego image will exactly look like the input cover image. Masking and filtering methods are restricted to 24 bit and gray scale images. In lossy compression, watermarking techniques can be applied without destructing the image. Masking technique is much better than LSB insertion if factors such as cropping, compressing and image processing are taken into consideration. In algorithms and transformations, the data in its frequency domain is hidden into transformed coefficients of the pixel values of the cover image. This provides high embedding capacity and resists malicious attacks. Some of the transform domain techniques are Discrete Wavelet transform (DWT), Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT).

From the study and analysis of different steganographic techniques, it is found that LSB replacement is the most simple and efficient technique for data hiding. In LSB replacement, in order to hide the data, only the least significant bits of pixel information are changing. Since they are the redundant bits, it does not affect the pixel information much if altered. This will avoid visual artifacts and distortions in the cover media where data is hidden.

4. STEGANOGRAPHY IN SYNTHESIZED TEXTURE

Otori and Kuriyamma[19] introduced the data hiding technique in texture synthesis. For that, they used pixel based texture synthesis and data hidden in the synthesized texture. In this method they conceal data in dotted patterns and then they are directly painted on the blank image as shown in figure 4.1. The final output texture is generated by performing the pixel based texture synthesis in the above image. Thus every pixels are generated until complete texture is obtained. Finally the dotted patterns get masked in the synthesized texture. For extracting the data, the print out of the stego image is taken before the data detection mechanism. The capacity of this method depends on the number of dotted patterns used for data hiding. However, this method has a small rate of error in data retrieval.

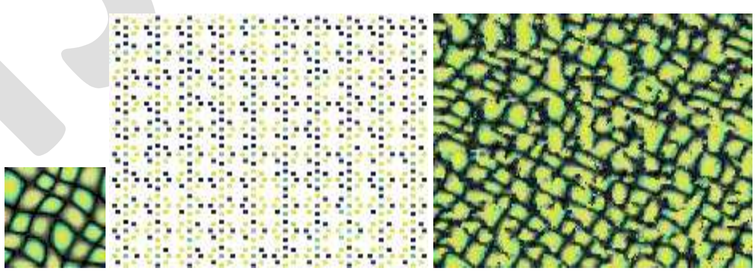

Fig 4.1: Input image, dotted pattern and the synthesized texture[19]

In the paper titled Robust data hiding on texture images by Otori and Kuriyamma[20], a regularly arranged dotted patterns painted with the color features that of the input image is hidden with the data. These dotted patterns are finally masked and this is having high quality compared to the existing texture synthesis methods. The hidden data can be recovered from the image printed like 2D bar codes. They also used the pixel based texture synthesis. The colors in the input image are grouped with a specific color component which is known as a coded component. The component value depends on the brightness of a particular position. A threshold level is also set in order to improve the robustness.

A new reversible data hiding scheme is put forward by Kuo-Chen Wu and Chung-Ming Wang[21] which used a patch based texture synthesis algorithm and yielded a better output. Initially an index table is generated from the secret key which records the location to paste the source patches. By pasting the source patches, in the preferred locations, a composition image is generated. Then, patch based texture synthesis method is performed in the composition image. The data is hidden in the image which is synthesized by using LSB replacement technique and is transmitted. At the receiver side, proper recovery steps are carried out. The important advantages of this method are that the data embedding capacity is increased and a normal steganalytic algorithm cannot defeat this approach. Also the source texture i.e, the input image can be perfectly retrieved without any distortion and the source texture can be again used for another round of steganography.

5. CONCLUSION

Steganography uses different approaches for the secure communication of information. Many different algorithms are combined in order to improve the efficiency and capacity of the data hiding methods. The most simple and one of the efficient methods is the LSB insertion technique. Steganography together with the texture synthesis methods is explained here. Texture synthesis process resamples an input texture and produces an output texture which is having a user specified size and having more data embedding capacity. From the review of different texture synthesis methods, it is found that patch based texture synthesis is the most fastest and efficient method for synthesizing a texture. So it is widely used in applications such as 3D Graphics, computer vision and animation industry. Due to the simplicity and effectiveness, LSB replacement is considered to be the better steganographic technique.

REFERENCES:

[1] 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.

[2] T. Morkel, J.H.P. Eloff, M.S. Olivier, “An Overview Of Image Steganography”, Information and Computer Security Architecture (ICSA) Research Group Department of Computer Science.

[3] Amruta G. Chakkarwar,” The Novel Approach To Enhance Image Using Texture Synthesis”, International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 4, April – 2013 ,ISSN: 2278-0181.

[4] Venkata Keerthy S, Rhishi Kishore T K,Karthikeyan B,Vaithiyanathan V,Anishin Raj M M, “A Hybrid Technique for Quadrant Based Data Hiding Using Huffman Coding”, IEEE Sponsored 2nd International Conference on Innovations in Information Embedded and Communication Systems ICIIECS'15, 2015.

[5] 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.

[6] Weiming Dong, Ning Zhou, Jean-Claude Paul, “Optimized Tile-Based Texture Synthesis”.

[7] Alexei A. Efros and Thomas K. Leung, ”Texture synthesis by non-parametric sampling”, IEEE International Conference on Computer Vision, pages 1033–1038, Corfu, Greece, September 1999.

[8] 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.

[9] Cosman, P.C, Oehler, K.L, Riskin, E.A, Gray, R.M, “Using Vector Quantization For Image Processing”, Proceedings of the IEEE, Sep 1993.

[10] David J. Heeger and James R. Bergen,”Pyramid-based texture analysis/synthesis”SIGGRAPH’95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, pages 229–238, New York, NY, USA, 1995. ACM.

[11] Peter J. Burt, Edward H. Adelson, “The Laplacian Pyramid as a Compact Image Code”, IEEE Transactions On Communications, Vol. Com-3l, No. 4, April 1983.

[12] Liu-yuan Lai, Wen-Liang Hwang, and Paruvelli Sreedevi, “Texture Synthesis Technique Using Different Sized Patches”, Institute of Information Science, Academia Sinica, Taiwan.

[13] 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.

[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] Zhu, S. C.,Wu, Y, And Mumford, D. 1998.” Filters, random-fields and maximum-entropy (Frame)”, Int. J. Comput. Vis. 27, 2 (March), 107–126.

[16] Renjie Chen, Ligang Liu, Guangchang Dong, " Local resampling for patch-based texture synthesis in vector fields", Int. J. of Computer Applications in Technology ,2004.

[17] 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.

[18] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," Computer, vol. 31, no. 2, pp. 26–34, 1998.

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

[20] Hirofumi Otori and Shigeru Kuriyama, "Robust Data Hiding on Texture Images", 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Toyohashi University of Technology.

[21] Kuo-Chen Wu and Chung-Ming Wang, "Steganography Using Reversible Texture Synthesis", IEEE Transactions On Image Processing, Vol. 24, No. 1, January 2015