ENHANCED STEGANOGRAPHY USING REVERSIBLE TEXTURE SYNTHESIS & MULTILAYER EMBEDDING TECHNIQUE

R. L. Sable

M.E. Student MSS's CET Jalna.

ABSTRACT

It is a unique approach for steganography using a reversible texture synthesis where rather than using existing image the algorithm will create new texture image using source texture and embeds message by texture synthesis process. The new image with similar appearance can be re-created by using texture synthesis process. The size of new image can be user specified. The important concept behind this kind of steganography is texture synthesis process. We can extract the embedded message and original texture image from final stego synthetic texture image. This new technique of steganography using reversible texture synthesis provide some advantages like the embedding capacity is directly proportional to the size of stego image. There will be no image distortion since size of new texture image is user specified. Also it provides the ability to recover source texture from the stego texture image. We can create visually plausible texture images by using this technique.

***Keywords*: Steganography, Data Embedding, Texture Synthesis.**

I INTRODUCTION

The meaning of Steganography is to hide text, audio, video behind the image. The main purpose of steganography is to hide information in such a way that attacker cannot detect hidden messages. The application of steganography includes a secure conversion of communication between two parties whose existence is unknown to an attacker and their success depends on detecting the existence of this communication [1].In a stenographic system, the information-hiding process is embedding process replaces these redundant bits with data from the hidden message to form a stego medium.[2] The goal of steganography is to keep the secret message undetectably. Most stenographic methods take over an existing image as a cover medium. When embedding secret messages into this cover image, distortion of image may occur. Because of this reason two drawbacks occur .First, the size of the cover image is fixed, so more secret messages are embedded allow for more image distortion. Therefore to maintain image quality it will provide limited embedding capacity to any specific cover image. Second, that image steganalysis approach is used to detect hidden messages in the stego image. This approach can defeat the image steganography and reveals that a hidden message is being carried in a stego image.

II RELATED WORK

J. Fridrich, M. Goljan, and R. Du [3] proposed a scheme for detecting least significant bit (LSB) non sequential embedding in digital images. The length of secret message is derived by examine the lossless capacity in the LSB and shifted LSB plane. The method analyzing lossless data embedding capacity in the LSBs. Randomizing the LSBs in the decreasing order of lossless capacity in the LSB Plane. Thus, the lossless capacity used to measure the degree of randomization of the LSB plane. Note that, the LSB plane of most images is random and it does not have any easily recognizable structure. To capture the degree of randomization using classical statistical quantities constrained to the LSB plane is unreliable. The lossless capacity revolves the fact that the LSB plane is related nonetheless to the other bit planes. Even if this relationship is nonlinear, the lossless capacity seems to measure this relationship equitably well. So, this method is used for steganography detection. In this technique, can be reliably detect the presence of secret messages in the images. M. F. Cohen [4] has used to implement an interactive application for texture design and synthesis. Texture is an image which has locality and stochastic property. Locality means small part of the image is look alike and they never look exactly the same (Stochastic). To overcome the memory consumption problems of large images generates a technique for tiling small images to fill a large area. We often need large texture images. So, we need to create large image from small samples .just tiling the samples is not a good method. Wang tile method is used or tile the plane with appropriate samples based on matching colors of adjacent edges.

If the set of tiles is rich enough and there is no periodicity, we can fill inside the tiles anything we want such as texture, geometric primitives etc. Using this method the user can fill Wang tiles on her own. The system interactively displays the result of the tiling. Using Wang Tiles method, once the tiles are filled, can be creates large expanses of non-periodic texture as needed very efficiently at runtime. Wang Tiles are squares shaped and each edge is has a color. A valid tiling requires matching colors to all shared edges between tiles. Another advantage is that, using a small set of tiles created from sample patches of a source image a highly compact representation for texture is achieved. If the two source images contain distributions of differing densities allows the creation of less uniform textures using two source images. An efficient real-time rendering of complex scenes can be done with modern graphics hardware by combining Wang Tiles with Layered Depth Images.

L. Liang, C. Liu[5] presented an algorithm for synthesizing textures from an input sample. This patch-based sampling algorithm is very fast and it creates high-quality texture image. This algorithm works well for a wide variety textures like's regular to stochastic textures. Can be sampling patches using a nonparametric estimation of the local conditional MRF density function .Also avoid mismatching features across patch boundaries of an image. The building blocks of the patch-based sampling algorithm are patches of the input sample texture to construct the synthesized texture. We can carefully select these patches of the input sample texture and paste it into the synthesized texture to avoid mismatching features across patch boundaries. Patch-based sampling algorithm combines the nonparametric sampling and patch pasting strengths .The texture patches in the sampling scheme provide implicit constraints to avoid garbage found in some textures.

A. A. Efros and W. T. Freeman [6] proposed a method for generates a new image by stitching together small patches of existing images. This process is known as image quilting. It is very fast and simple texture synthesis algorithm. By extend this algorithm to perform texture transfer operation. In patch-based texture synthesis procedure, define the square block of user-specified size from the set of all such overlapping blocks in the input texture image. To synthesize a new texture image, let us simply tile the blocks taken randomly from the input texture image. Next step is to introduce some overlap in the placement of blocks onto the new image. Now, search source texture for such a block that agrees some measure with its neighbors along the region of overlap .At last, let the blocks have ragged edges which will allow them to better approximate the features in the texture. Before placing the block into the texture can be calculates error in the overlap region between it and the other blocks. Then find a minimum cost path through that error surface and find boundary of the new block.

Z. Ni, Y.-Q. Shi [7] presented a reversible data hiding algorithm for recover the original image without any distortion from the marked image after the hidden data have been extracted. The zero or the minimum points of the histogram of an image is utilized by this algorithm and slightly modifies the pixel grayscale values for embedding the data into the image. By comparing the existing reversible data hiding algorithms, It can embed more data. The algorithm applicable to a wide range of images such as commonly used images, medical images, texture images, aerial images and all of the 1096 images in CorelDraw database. This method can embedded a large amount of data at the same time keeping a very high visual quality for all natural images, specifically, the PSNR of the marked image versus the original image is guaranteed to be higher than 48 dB. This technique is applicable to all types of images. This proposed lossless data hiding technique is applied to still images and videos. Videos consist of a sequence of images. In the proposed algorithm, embedding and extracting of data are presented in terms of pseudo code.

X. Li, B. Li, B. Yang, and T. Zeng[8], have used a Histogram shifting (HS) technique for reversible data hiding (RDH). Using HS-based RDH method, high capacity and low distortion can be achieved. This paper presents a general framework to construct HS-based RDH technique. Using this proposed framework, can be get a RDH algorithm by simply designing shifting and embedding functions. In this method, first divides the host image into nonoverlapping blocks such that each block contains n pixels. Then, generates an n-dimensional histogram by counting the frequency of the pixel-value-array sized n of each divided block. At last, modifies the resulting n-dimensional histogram for implementing the data embedding scheme.

H. Otori and S. Kuriyama [9] have presented a data hiding techniques tools for protecting copyright or sending secret messages. This paper proposed a method for embedding arbitrary data by synthesizing texture images using the smart techniques of generating repetitive texture patterns through feature learning of a sample image. By extending this technique, a synthesized image can effectively conceal the embedded pattern, and the pattern can be robustly detected from a photographed image. This method introduces a random coating and re-coating to improve the quality of the texture image synthesized from the initial painting of LBP. The algorithm focuses on the textures that are iteratively generated by learning a pattern of an exemplar. This is infeasible for a procedurally and randomly generated pattern. Computes the shape of a histogram of the LBPs for every pixel inside a divided image block and embedded the data onto it. The implementation requires border lines for extracting the square region of a data-embedded area in the texture image. This is done by developing a technique of automatically determining the square region. This paper uses texture images for embedding data because texture patterns are widely utilized artificial images. Texture images can be automatically generated by computing the feature of the iterative patterns of real objects. So that we can embed data by affixing a seal of a synthesized image on a real object in unnoticeable manner, with quality of the anteed. We directly paint the data by converting its numerical value into a dotted colored pattern rather than changing the color component of images .Then automatically coat a texture image onto the painted pattern from a sample image (or exemplar) so as to conceal its existence with a natural texture pattern.

Y. Guo, G. Zhao[10] proposes a video texture synthesis method. Two key factors, such as frame representation and blending artifacts, that affect the synthesis performance. To improve the synthesis performance from two features. First, effective frame representation is used to capture both the longitudinal information in temporal domain and the image appearance information in spatial domain. Second, Artifacts that reduce the synthesis quality are significantly suppressed on the basis of a diffeomorphic growth model. The proposed video texture synthesis approach has mainly two stages such as video stitching stage and transition smoothing stage. In the video stitching stage, a video texture synthesis model is proposed to generate an infinite video flow. This paper presents a new spatial temporal descriptor to gives an effective representation for different types of dynamic textures. In the second stage of video synthesis, a smoothing method is presented to improve synthesis quality. It aims to set up a diffeomorphic growth model to emulate local dynamics around stitched frames. First, a Multi-frame LBPTOP frame signature is proposed to capture both the spatial and temporal information. Based on this frame signature, it identifies the most appropriate matching pairs of frames. Second, a diffeomorphic growth model is applied to identify matching frames. For smoother transition, this growth model can emulate temporal motion around matching frames and estimate virtual frames. This synthesis method has some advantages .First, combines the spatial and temporal description in each feature, which enhances the ability of capturing the relationships among neighboring frames. Second, considers abundant temporal discriminative information, which makes it flexibly adaptive to dynamic textures with different properties. Third, gets more natural visual effects by using the diffeomorphic growth model.

A. A. Efros [11] presented a non-parametric method for texture synthesis. The texture synthesis process emerges a new image outward from an initial seed; consider one pixel at a time. The objective of this method is to preserve local structure and produces good results for a wide variety of synthetic and real-world textures. The algorithm considers texture, pixel by pixel, outwards from an initial seed. First, chose a single pixel so that the model captures high frequency information as possible. Using probability tables for the distribution of single pixel can be synthesis the process by using all possible contexts. An approximation can be getting by using various clustering techniques. This method generates texture as a Markov Random Field (MRF). It assumes the probability distribution of brightness values for a pixel given the brightness values of its spatial neighborhood and the rest of the images are independent. The neighborhood of a pixel is design as a square window around that pixel.

R. Rejani[12] have proposed a pixel pattern based steganography. This method involved hiding the message within in an image by using the existing RGB values whenever possible at pixel level or with minimum changes. A key is present along with the image, which can be used to decrypt the message stored at pixel levels. This method presents an improved steganography technique for embedding secret message bit in image metadata fields based on the RGB values and the position of the pixels. The pixels in the image changed only for characters and the algorithm cannot find a pixel which can represent it. Because only the metadata is modified, the stego image looks exactly the same as original image. In this method, only the size of the stego image will increased. Most of the existing image steganographic algorithms have many drawbacks. First, the size of the cover image is fixed, so embedding more secret messages will leads to image distortion. So it needs a compromise between the embedding capacity and the image quality, which results in the limited capacity provided in any specific cover image. Second, that image steganalytic algorithm can be used to detect secret messages hidden in the stego image. To overcome these limitations, Kuo-Chen

Wu and Chung-Ming Wang [13] have proposed an approach for steganography using a reversible texture synthesis. A texture synthesis process synthesizes a new texture image from a smaller texture image which has a similar local appearance and an arbitrary size. This method combines the texture synthesis process with steganography to conceal secret messages.

III PROPOSED WORK

We propose a novel approach for steganography using reversible texture synthesis A texture synthesis process resample's a small texture image and provides an image of arbitrary size and shape, which holds the hidden message. It involves generation of an index table with a desirable number of rows and columns. The index table is split into given number of rows and columns and the secret message is hidden in any of the column or row according to the form with a merging file name. The image will only be sent to the list of available users, that is the users with valid username and registered users. At the receiver end, the receiver, for whom the message is intended, will receive it in their inbox. As the splitting procedure is only revealed to the registered user, they split the merged image using secret key and read the message. First, we will define some basic terminology to be used in our algorithm. The basic unit used for our steganographic texture synthesis is referred to as patch.
The proposed method is described as follows. The basic unit of the steganographic texture synthesis is introduced to as a texture where its size is user-specified. The patches are combined together to form the composition image in which we are embedding our secret message. The project includes mainly three major steps.

1) Message Embedding Procedure
2) Source Texture Recovery, Message Extraction and Message Authentication Procedure
3) Capacity Determination

1**. Concepts involved in Message Embedding Procedure** The message embedding procedure involve mainly four steps. They are

A. Index Table Generation
B. Patch Composition Process
C. Message Oriented Texture Synthesis Generation.
D. Multilayer Embedding technique

A. Index Table Generation

The first process of this project is the index table generation where here will create an index table to reserve the location of the source patch set inside the synthetic texture. The index table will allow us to access the synthetic texture and extract the source texture wholly. The texture of any size according to our wish can be generated using this index table.

B. Patch Based Composition

The second step that has to be used in this project is to attach the source patches into a workbench to create a composition image. First here will set up an empty image as the workbench where the size of the workbench is proportional to the synthetic texture. By referring to the source patch IDs stored in the index table, we then attach the source patches into the workbench. During the attaching process, if no imbrications of the source patches are found, we can attach the source patches directly into the workbench.

C. Message Oriented Texture Synthesis Generation.

After the creation of the composition image we have to embed the secret message through the message oriented texture synthesis to generate the final steno synthetic texture.

2. Source Texture Recovery, Message Extraction, and Message Authentication Procedure the message extracted for the receiver side consist of creating the index table, attaining the source texture, performing the texture synthesis, and extracting and authenticating the secret message hidden inside the stego synthetic texture.

3. Capacity Determination.

The next step is to look for how much data can be embedded in the stego texture image. The embedding capacity can be related to the capacity in bits that can be hidden at each patch (BPP, bit per patch), and to the number of embeddable patches in the stego synthetic texture (EPn). Each patch can hide at least one bit of the secret message.

D. Multilayer Embedding technique

To improve the embedded capacity and image quality we used this multiple layer embedding technique. To implement this we can use a simple scheme like Least Significant Bit (LSB) embedding. The technique works by replacing some of the information in a given pixel with information from the data in the image. When we try to hide our data in a texture then we require enough budget of LSBs to hide our data in. These bits are located in the image pixels. Each pixel has three elements (R, G, and B) that represent the Red, Green, and Blue color elements of the pixel by assuming non-transparent image; these elements can have a value between 0 and 255.Suppose we have an image having pixel width 300 and 400 pixels height, then we'll have 300 x 400 x 3 = 360000 LSBs. Each character can be represented by 8 bits, then that image can hide 360000 / 8 = 45000 characters.

The method of steganography using reversible texture synthesis is mainly used for hide the secret messages. A new texture image is synthesizes from several tiny texture images by using the texture synthesis process. The method consists of combination of both texture synthesis process and steganography. It contains mainly two procedures [1]. 1: Message embedding procedure 2: Message extracting procedure In message embedding procedure, the first procedure is dividing the source texture image into different image block. This image block is called as patches. To table is used. The workbench is blank image whose size is same as that of synthetic texture. With the help of source patch ID which is placed in the index table, the corresponding source patches are paste into the workbench to generate a composite image. After pasting the source patch the next step is to find mean square error (MSE) of overlapped region. This overlapped area is found in between the patch which we want to insert in the workbench and the synthesis area. The resultant patches are ranked as per the ascending order of mean square error (MSE). And finally the patches are selected from given list in such way that the rank of patches is equals to decimal value. The decimal value is nothing but the n-bit value of our secret message. At receiver side, the index table is generated by using secrete key which the receiver already have. To retrieve the size of the expected source texture we can refer each patch region and its related order which is present in the index table. After retrieving the size the blocks are arranged as per their corresponding order. Next step is authentication - We were going to suppose the present working location of workbench and similarly the working location of stego synthetic texture to predict the stego block region. The stego block region is used to search candidate list and to check whether there is any patch from candidate list having similar kernel region as the corresponding stego block region. If such similar patch is found, The rank is given to this matched patch. We can represent the value of secrete bits in patch which is in decimal format. This process is called as message extracting procedure.

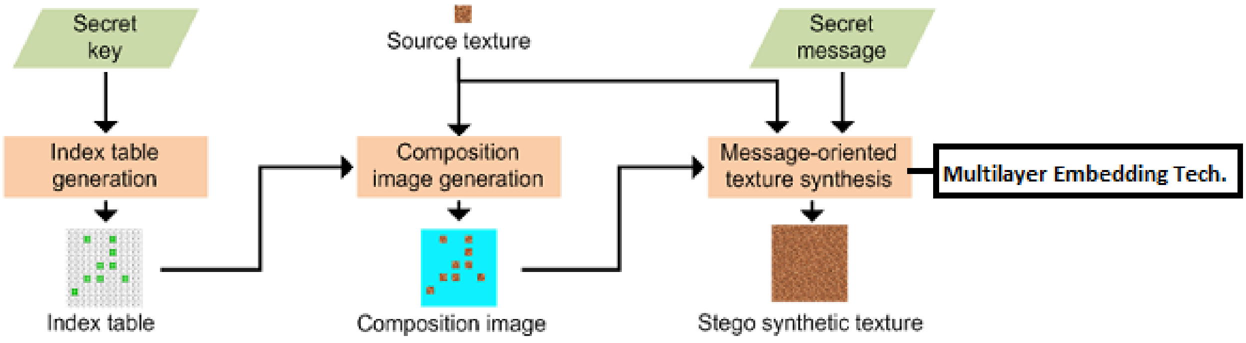

Fig. (a)Message Embedding Procedure

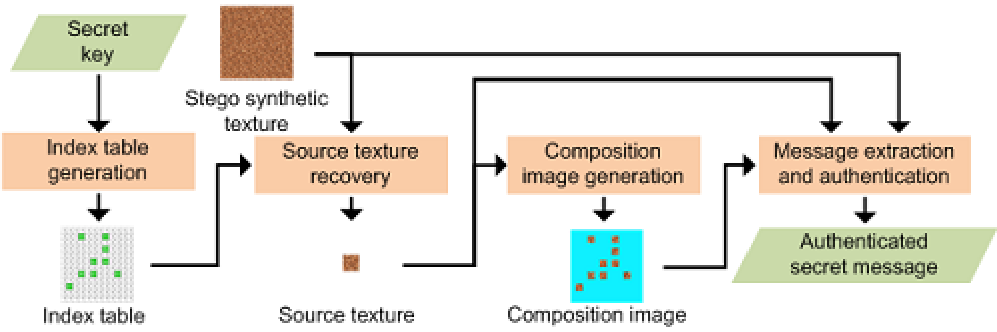

Fig. (b) Message Extracting Procedure

IV ALGORITHM HIDING THE TEXT INSIDE THE IMAGE

1. Loop through the pixels of the image. In each iteration, get the RGB values separated each in a separate integer.
2. For each of R, G, and B, make the LSB equals to 0. These bits will be used in hiding characters.
3. Get the current character and convert it to integer. Then hide its 8 bits in R1, G1, B1, R2, G2, B2, R3, G3, where the numbers refer to the numbers of the pixels. In each LSB of these elements (from R1 to G3), hide the bits of the character consecutively.
4. When the 8 bits of the character are processed, jump to the next character, and repeat the process until the whole text is processed. 5. The text can be hidden in a small part of the image according to the length of that text. So, there must be something to indicate that here we reached the end of the text. The indicator is simply 8 consecutive zeros. This will be needed when extracting the text from the image.

Extracting the text from image

It's simpler than hiding. Just pass through the pixels of the image until you find 8 consecutive zeros. As you are passing, pick the LSB from each pixel element (R, G, B) and attach it into an empty lue. When the 8 bits of this value are done, convert it back to character, and then add that character to the result text you are seeking.

V RESULTS

Results of Embedding Capacity On personal computer we collect our experimental results. For results we can take the four source texture. Following table shows the embedding capacity of our algorithm when various resolutions of synthetic texture are produced. For fixed number of BPP, the larger the resolution of source texture S_w x S_h (96x96 vs.192x192). Overall embedding capacity is TC. For 10 BPP our procedure will offer 6160 bits vs. 5890 bits. Because the large source texture contains more source patches S (9 vs. 36) and we need to paste these source patches which S_{pn} cannot hide any secret bits. It will decrease the number of embeddable patches E . Following graph presents the total E_{en} embedding capacity our algorithm. However, we can hire BPP (11 vs. 14) for conveying the further secret messages (6776bits vs. 8246 bits). The maximal capacity provided by our algorithm is 36456 bits provide. Several techniques can be used for determining the image quality of stego synthetic texture. To determine the image quality of synthetic results we define dimension called as mean squared error of the overlapped area (MSEO). This MSEO will reveal the similarity between the candidate patch and the synthesized area where can activate the image quilting technique during message oriented texture synthesis process. During the texture synthesis process we can calculate MSEO.

TOTAL EMBEDDING CAPACITY IN BITS			
Synthetic texture size: $T_w \times T_h$ =1008×1008; Patch size: P_w*P_h=48×48: Boundary depth: P =8			
$S_w \times S_h$	SP_n	EP_n	TC(5BPP)
96×96	9	616	3150
128×128	16	609	3100
192×192	36	589	3050

Table: ***Embedding Capacity***

VI CONCLUSION

This paper advises a reversible steganographic algorithm using texture synthesis with multilayer embedding. This system produces a large stego synthetic texture which can hide secret messages. Thus we can say that our proposed mode allows the users to perform multiple layers embedding to achieve very high embedding capacity and the good quality of stego images. We can conclude that our offered method is applicable for some information hiding applications such as secret communication, medical imaging systems, and online content scattering systems.

ACKNOWLEDGMENT

It is pleasant Endeavour to present Transaction paper report on "**Enhanced Steganography Using Reversible Texture Synthesis & Multilayer Embedding Technique"** I take this opportunity to express my gratitude towards my guide **Prof. Ms. S. G. Tuppad** for her constant encouragement and guidance. He is a constant source of motivation and inspiration. Without his efforts the project could not have taken final phase.

I would also like to thank to **Prof. Ruhi Sodani** and **Dr. C. M. Sedani** for their cordial support who have co-operated and provided valuable information to complete this Project. I would like to extend my gratitude to all my friends, who have extended their support and helped me to complete Project Report. Words fall short to express my deep sense of gratitude towards them all, who have directly or indirectly helped in making this project.

REFERENCES

[1] 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.

[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] 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, Oct./Dec. 2001.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] X. Li, B. Li, B. Yang, and T. Zeng, ―General framework to histogram shifting- based reversible data hiding,‖ *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2181–2191, Jun. 2013.

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

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

[11] 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.

[12] R. Rejani,D. Murugan and Deepu V. Krishna―Pixel pattern based steganography on images‖ *journal on image and video processing*, feb 2015, volume: 05, issue: 03

[13] Kuo-Chen Wu and Chung-Ming Wang―Steganography Using Reversible Texture Synthesis ‖ *IEEE Trans .on* Image Processing, VOL. 24, NO. 1, Jan 2015