Secret Data Embedding using Texture Synthesis
Suphiya P. Inamdar, Suhas B. Bhagate
Department of Computer Science and Engineering, D.K.T.E. Society’s Textile and Engineering Institute, Ichalkaranji, Maharashtra, India

Abstract—A steganography is an art of hiding confidential data into digital media such as image, audio, video etc. The proposed system uses steganography using reversible texture synthesis. Texture synthesis uses the concept of patch which represents an image block of source texture where its size is user specified. A texture synthesis process resamples a smaller texture image, and provides a new image with arbitrary size and shape. Instead of using an existing cover image to hide messages, the algorithm conceals the source texture image and embeds secret messages using the process of texture synthesis. This allows extracting the secret messages and source texture from a stego synthetic texture. The approach offers some advantages. First, the scheme offers the embedding capacity that is proportional to the size of the stego texture image. Second, the reversible capability inherited from this scheme provides functionality, which allows recovery of the source texture. And third, there will be no image distortion since size of new texture image is user specified.

Keywords—Steganography, Data embedding, Texture synthesis, Cover medium, Index table

I. INTRODUCTION

Texture synthesis is the process of re-samples a small texture image drawn by an artist or captured in a photograph in order to synthesize a new texture image, which have a similar local appearance and arbitrary size. This project aims to combine the texture synthesis process into steganography [1] to conceal secret messages as well as the source texture. The secret messages and the source texture can be extracted from a stego synthetic texture.

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. 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 can be improved.

In texture synthesis process pixel-based algorithms generate the synthesized image pixel by pixel and use spatial neighbourhood comparisons to choose the most similar pixel in a sample texture as the output pixel. This output pixel is determined by the already synthesized pixels, any wrongly synthesized pixels during the process influence the rest of the result causing propagation of errors. The steganography and reversible texture synthesis is based on patch-based algorithms which paste patches from a source texture instead of a pixel to synthesize textures. This method improves the image quality of pixel-based synthetic textures because texture structures inside the patches are maintained. A patch represents an image block of a source texture where its size is user-specified. This algorithm conceals the source texture image and embeds secret messages through the process of texture synthesis. This is to extract the secret messages and source texture from a stego synthetic texture.

II. LITERATURE SURVEY

A simple and efficient pixel based method is presented by Efros and Leung [3] in 1999. This method generates the synthesized image pixel by pixel and use spatial neighborhood comparisons to choose the most similar pixel in a sample texture as the output pixel. 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. But any wrongly synthesized pixels during the process influence the rest of the result causing propagation of errors.

Wei and Levoy [4] presented 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) 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, leaves, pebbles etc.
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 of textures like 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 generating 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.

Liang, Liu and et al.[7] 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 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. 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.

A. A. Efros and W. T. Freeman [6] proposed a method for generating 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 [8] 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 gray scale values for embedding the data into the image. By comparing the existing reversible data hiding algorithms [9], 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 techniques is applicable to all types of images. This proposed lossless data hiding technique is applied to still images and videos.

X. Li, B. Li, B. Yang, and T. Zeng [10], 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...
In this method, first divides the host image into no overlapping 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.

III. METHODOLOGY

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 [11].

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 record the corresponding source patch’s location the index 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.

![Message embedding procedure](image1)

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.

![Message extracting procedure](image2)
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 secret bits in patch which is in decimal format. This process is called as message extracting procedure.

IV. CONCLUSIONS

This paper proposes a reversible steganographic algorithm using texture synthesis. Texture Synthesis method provides reversibility to retrieve the original source texture from the stego synthetic textures and it can be used for second round of texture synthesis if needed. Studies show that the previous methods have many drawbacks; this can be solved using the new Steganography method using Reversible texture synthesis. Our method can achieve reversibility, separate data extraction and image recovery.

REFERENCES