Enhancing Security of Image by Steganography using Reversible Texture Synthesis

Aparna Vinod
Department of Computer Engineering
College of Engineering Chengannur,
Alappuzha, India

Sony P
Assistant Professor
Department of Computer Engineering
College of Engineering Chengannur,
Alappuzha, India

***Abstract*— Steganography is a method of information hiding. In this paper, introduce a novel approach for enhancing the security of an image by steganography using reversible texture synthesis. The texture synthesis process produces an arbitrary large and similar looking new texture from a small texture image. This algorithm conceals the source texture and embeds the secret image through the process of texture synthesis. So we can extract the secret image and the source texture from a stego synthetic texture. This method offers four distinct advantages. First, the embedding capacity is proportional to the size of the stego synthetic texture image. Second, it is very difficult for a steganalytic algorithm to defeat this steganography method. Third, recovery of the source texture from the stego image is done. Fourth, prevent loss of information due to the mirroring operation. The proposed algorithm is secure and robust against an RS steganalysis attack.**

***Keywords*— Data embedding, reversible, steganography, texture synthesis.**

I. INTRODUCTION

Digital technology gives us new ways to apply steganographic techniques, including one of the most intriguing-that of hiding information in digital images. Steganography is the art and science of hiding communication. It hides the very existence of the message by embedding it inside a carrier file of some type. The most important requirement of steganography is detectability; the concealed messages should be perfectly disguised under all statistical and visual analysis. A typical steganographic application includes covert communications between two parties whose existence is unknown to a possible attacker and whose success depends on detecting the existence of this communication [1]. The main categories of file formats that can be used for steganography are: Text Steganography, Image steganography, Audio steganography, and Protocol steganography. A large number of image steganographic algorithms have been investigated with the increasing popularity and use of digital images.

Most of the image steganographic algorithms use an existing image as the cover medium. So embedding secret image into this cover image will result image distortion in the stego image. This leads to two drawbacks. First, size of the cover image is fixed, if add more data then image distortion will occur. Consequently, there is a compromise reached between the embedding capacity and the image quality. Second, If a stego image contains some distortion, and regardless of how minute it is, this will interfere with the natural features of the cover image. Image steganalysis is an approach used to detect secret messages hidden in the stego image. An image steganalytic algorithm can easily defeat the image steganography and then reveal the hidden image is being conveyed in a stego image.

To overcome the above drawbacks, a new approach for steganography using reversible texture synthesis is proposed. A texture synthesis process produces an arbitrary large and similar looking new texture from a small texture image. Weave the texture synthesis process into steganography concealing secret messages as well as the source texture. Algorithm conceals the source texture and embeds the secret image through the process of texture synthesis. So we can extract the secret messages and the source texture from a stego synthetic texture. Steganography also takes advantage of the reversibility through the texture synthesis.

This approach offers four advantages. First, due to the capability of texture synthesis method, it can synthesize an arbitrary size of texture images. Thus embedding capacity is proportional to the size of the stego texture image. Secondly, it is very difficult for a steganalytic algorithm to defeat this steganography method. Since the stego texture image is composed of a source texture instead of modifying the existing image contents. Third, the reversible capability. Because we can recover source texture from stego synthetic texture. Recovered source texture is exactly the same as the original source texture. It can be also used for the steganography of the second round of secret message if needed. This algorithm can provide a wide range of embedding capacities, produce visually believable texture images, and also recover the source texture.

II. LITERATURE SURVEY

Steganography is the art of hiding secret information inside an image. The purpose of steganography is covert communication to hide the existence of a message from a third party. The major applications of the system included in military area and banking.

Reversible data embedding algorithm [2], which embed a large amount of data while keeping a very high visual quality for all natural images. Recover original image from the marked image after the hidden data have been extracted. It utilizes the zero or the minimum point of the histogram and slightly modifies the pixel gray scale values to embed data.

The number of bits that can be embedded into an image equals to the number of pixels which are associated with the peak point. The computation is quite simple and the execution time is rather short.

In Line-Based Cubism method [3] combining art image generation and data hiding to enhance the camouflage effect for various information hiding applications. Data hiding with the minimal distortion is carried out during the process of recolouring the regions in the generated art image by shifting the pixels' colours for the minimum amount of ±1 while keeping the average colours of the regions unchanged. Also attract people by the artistic content of the Cubism-like image, gives the data hiding technique a camouflage effect which arouses no suspicion from hackers. Data hiding technique is very suitable for use in covert communication or secret keeping.

In Reversible Data Hiding in Encrypted Image [4] method provide novel reversible data hiding scheme for encrypted image, which is made up of image encryption, data embedding and data-extraction image-recovery phases. The data of original image are entirely encrypted by a stream cipher. According to the data-hiding key, with the aid of spatial correlation in natural image, the embedded data can be correctly extracted while the original image can be perfectly recovered.

The Reversible watermarking of [5] is very efficient. Here investigates the use of local LS prediction in DE reversible watermarking. The basic idea is to compute, for each pixel, a distinct LS predictor on a block centred on the pixel. The same predictor is recovered at detection, avoiding the need of embedding a large amount of additional information. For the actual cases of least square predictors with identical context because the median edge detector, gradient-adjusted predictor or the easy parallelogram neighbourhood, the native prediction-based reversible watermarking clearly outperforms the progressive schemes supported the classical counterparts.

The Histogram-Shifting-Based Reversible Data hiding [6] provide a general framework to construct HS-based RDH. According to the framework, to obtain a RDH algorithm, one just needs to define the shifting and embedding functions. This method will facilitate the design of RDH. The framework has a potential to provide excellent RDH algorithms.

III. SCOPE

Most of the steganographic algorithms adopt an existing image as cover medium. The embedding of secret message into the cover image can lead to image distortion in stego image. So image steganalytic algorithm can easily defeat the image steganography. For avoiding the above disadvantage introduce a novel algorithm that can provide various numbers of embedding capacities, produces reasonable texture images and recover the source texture. Reversible texture synthesis is the approach for retrieving secret image and source texture from stego image.

IV. METHODOLOGY

The basic unit used for the steganographic texture synthesis referred as a "patch". It represents an image block of a source texture. The size of the patch is user-specified. Patch contains the central part and an outer part. Central part is referred to as the kernel region with size of $K_w \times K_h$. The part surrounding the kernel region is referred to as the boundary region with the depth (P_d).

We take source texture. Size of the source texture is denoted by its width (S_w) and its height (S_h). Subdivide the source texture image into a number of nonoverlapped kernel blocks, each with the size of $K_w \times K_h$. Then expand a kernel block with the depth P_d at each side to produce a source patch. The expanding process will overlap its neighbour block.

A. *Image Embedding Procedure*

Image embedding involves index table generation, composition image generation, encryption and image oriented texture synthesis. Here we hide an image in a texture image through the process of texture synthesis. Steganalytic algorithm cannot find any difference in the stego image.

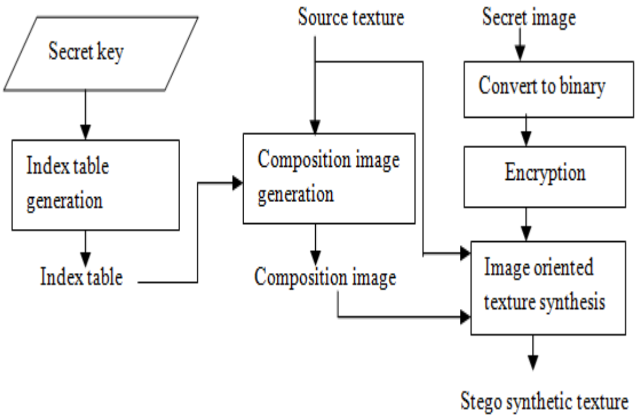

Fig. 1. Image embedding procedure.

1) *Index Table Generation Process*

The first process is the index table generation. Here we produce an index table to record the location of the source patch set *SP* in the synthetic texture. The index table allows us to access the synthetic texture and retrieve the source texture completely.

The index table has the initial values of −1 for each entry, which indicate that the table is blank. Then re-assign these values according to the source patch ID distribute in the synthetic texture. Use a random seed for patch ID distribution, which increases the security of the steganographic algorithm. It is difficult for steganalytic algorithm to extract the source texture. Because the index table will be scattered with different values. The entries with non-negative values indicate the corresponding source patch ID subdivided in the source texture. The entries with the value of −1 represent that the patch positions will be synthesized by referring to the secret message in the message-oriented texture synthesis. Entries with non negative values represent the source patch ID.

2) *Patch Composition Process*

Here we paste the source patches into a workbench to produce a composition image. For this, first establish a blank image as the workbench. The size of the workbench is same as the size of the synthetic texture. Then paste the source patches into the workbench according to the source patch IDs stored in the index table. During the pasting process, source patches directly paste into the workbench when there is no overlapping of the source patches is encountered. If any overlapping of source patches occur, then use the image quilting technique [6] to reduce the visual artefact on the overlapped area.

3) *Conversion*

The third process is conversion. Here take a secret image for embedding. If the chosen secret image is not binary, then convert the secret image into binary image for effectiveness of the image hiding.

4) *Encryption*

Here the binary image is encrypted using a key. Binary image can be encrypted by applying the bitwise XOR operator to every pixel using a given key. The key stream is generated by a pseudo-random number generator. Same key is used in the encryption and decryption phase.

5) *Image Oriented Texture Synthesis Process*

In this process, embed the secret image via image oriented texture synthesis to produce final stego image. Proposed method uses pixel based texture synthesis [7] rather than base paper texture synthesis process. In the base paper texture synthesis process generate a set of candidate patches from the source texture. Then compute the mean square error (MSE) of the overlapped region between the synthesized area and the candidate patch. After computing rank of all the candidate patches, further rank these candidate patches according to their MSEs. Once the ranks of all candidate patches are determined, select the candidate patch where its rank equals the decimal value of an *n*-bit secret message. In this way, a segment of the *n*-bit secret message has been concealed into the selected patch to be pasted into the working location. However, if pasting locations cause the source patches to overlap each other, we employ the image quilting technique [6] to reduce the visual artifact on the overlapped area.If a kernel block is located around the boundary of a source texture, operate the boundary mirroring using the kernel block's symmetric contents to produce the boundary region. Due to the mirroring operation, synthetic images can be broken down. By comparing the similarity of an expanded boundary area with an inside boundary area, search patches in the synthetic image and identify the patches coming from the four corners and boundaries of source image. It leads to loss of information. So it is insecure.

For gaining more security proposed method uses pixel based texture synthesis. This class of methods adopts the strategy of copying one pixel at a time. Loosely, it loops over the set of unknown pixels which are adjacent to known pixel values. For each unknown pixel in this set, it finds the set of all matching pixel neighbourhoods from the original Image. Use a Gaussian-weighted sum-of-squared-difference (SSD) score to measure the match between the template and a patch in the image. Arrange the matching neighbourhoods in ascending order of the SSD value. Then check the secret binary image pixel value. If the binary image value is 1 then chosen the first matching neighbourhood value to substitute unknown pixel otherwise choose the second matching neighbourhood. In this manner secret image embedded in the synthesised image. Embedding completed, but till some pixel have unknown value. In this case randomly picks a matching neighbour to fill in the current unknown pixel. It demonstrated power of probability sampling by synthesizing high quality results for a broad range of texture samples.

B. *Image Extraction and Authentication Procedure*

The message extracting for the receiver side involves generating the index table, retrieving the source texture, performing the texture synthesis, extracting and authenticating the secret image concealed in the stego synthetic texture and decryption to get hidden secret image.

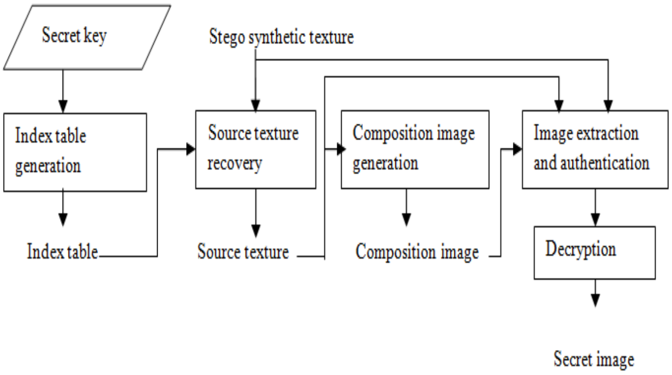

Fig. 2. Message extracting procedure.

Given the secret key held in the receiver side, the same ***index table*** as the embedding procedure can be generated.

Second process is source texture recovery. Each kernel region with the size of $K_w \times K_h$ and its corresponding order with respect to the size of $S_w{\times}S_h$ source texture can be retrieved by referring to the index table with the dimensions $T_{pw}{\times}T_{ph}$. Then arrange kernel blocks based on their order, thus retrieving the recovered source texture which will be exactly the same as the source texture.

In the composition image generation phase, paste the source patches into a workbench to produce a composition image by referring to the index table. This generates a composition image that is identical to the one produced in the embedding procedure.

In image extraction and authentication phase, same procedure in embedding is performed to find the matching neighbourhood of the unknown pixel from the source texture. For this calculate the SSD score. Arrange the matching neighbourhood in the ascending order of the SSD score. At the extraction and authentication phase, consider the unknown pixel in composition image. Then find a pixel which corresponds to the unknown pixel at the same position in synthetic texture. Then compare the pixel value with first matching neighbourhood, if they match then embedded bit is 1. Otherwise check the pixel value with second matching neighbourhood, if they match embedded bit is 0. Extract the embedded pixel value in this manner. However, if cannot disclose such a exact match it means that the stego kernel region has been tampered with, leading to a failure of the image authentication. In this way, authenticate and extract the secret image that is concealed in the stego synthetic texture pixel by pixel.

Choose the same key in encryption to decrypt the image. Image can be decrypted by reapplying the bitwise XOR operator to every pixel using a given key. The key stream is generated by a pseudo-random number generator. Appropriate modulation

V. CONCLUSION

Image steganography is the way of secret communication through the digital images. In this project introduces a novel approach for steganography using reversible texture synthesis. Given an original source texture, this scheme can produce a large stego synthetic texture concealing secret image. In this method, weave the steganography into a pixel based texture synthesis. The secret image is encrypted before it embedding into the synthetic texture. The texture synthesis process resamples a smaller texture image, which synthesizes a new texture image with a similar local appearance and an arbitrary size. It provides reversibility to retrieve the original source texture from the stego synthetic textures, making possible a second round of texture synthesis if needed. It offers some advantages such as the embedding capacity that is proportional to the size of the stego texture image, a steganalytic algorithm is not likely to defeat the steganographic approach, the reversible capability of texture synthesis provide the functionality to allows recovery of the source texture and avoid loss of information due to the boundary mirroring operation. The presented algorithm is secure and robust against an RS steganalysis attack.

VI. REFERENCES

[1] 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.

[2] S.-C. Liu and W.-H. Tsai, “Line-based cubism-like image—A new type of art image and its application to lossless data hiding,” IEEE Trans. Inf. Forensics Security, vol. 7, no. 5, pp. 1448–1458, Oct. 2012.

[3] Xinpeng Zhang,”Reversible Data Hiding in Encrypted Image, ” IEEE Signal Process Let., Vol. 18, no. 4, AprL 2011

[4] I.-C. Dragoi and D. Coltuc, “Local-prediction-based difference expansion reversible watermarking,” IEEE Trans. Image Process., vol. 23, no. 4, pp. 1779–1790, Apr. 2014.

[5] 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.

[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] A. A. Efros and T. K. Leung**, “** Texture synthesis by non-parametric sampling”, in Proc. 7th IEEE Int. Conf. Comput. Vis., 1033–1038, 1999.

[8] Kuo-Chen Wu and Chung-Ming Wang, ”Steganography Using Reversible Texture Synthesis”, IEEE Trans.ImageProcess, vol. 24, no. 1,Jan 2015.