International Journal of Computer Sciences (IJCS)
ISSN (Online): 4142-3453 Impact Factor: 3.5

CRYPTOGRAPHY VICTIMIZATION FITFUL FRAMEWORK INTEGRATION

B Neeraja[1] , S J Sowjanya[2], Y.Subba Rayudu[3]

[1]Assistant Professor ,Department of Computer Science & Engineering,Institute of Aeronautical Engineering ,Hyderabad

[2] Associative Professor ,Department of Computer Science & Engineering,Institute of Aeronautical Engineering ,Hyderabad

[3] Assistant Professor of Cse Deparment ,Institute of Aeronautical Engineering ,Hyd

Abstract-A new steganographic method using reversible texture synthesis is proposed in this paper. A texture synthesis process resamples a smaller texture image which synthesizes a new texture image with a similar local appearance and arbitrary size. Along with reversible texture synthesis process, reserving room method is used to embed additional data, which is the highlight of this paper. Traditional RDH algorithm is used for reserving room method. Data hider can reversibly embed additional data in stegosynthetic texture image. The proposed method can achieve real reversibility, that is, texture image recovery and data extraction cause no error.

Keywords: Data embedding, Reversible texture synthesis, Tree based parity check (TBPC)

1. INTRODUCTION

Steganography is the method of hiding a message, file, image, or video within another file, message, image, or video. The word steganography combines from the two Greek words “steganos” means “protected”, and “grapheins” means “writing”. The advantage of steganography than cryptography is that the secret message does not attract the attention of the attackers by simple observation. The cryptography protects only the content of the message, while steganography protects the both messages and communication environment.

In most of the image steganographic methods, uses the existing image as their cover medium. This leads to two drawbacks. Since the size of the cover image is fixed, embedding a large secret message will results in the distortion of the image. Thus a compromise should be made between the size of the image and the embedding capacity to improve the quality of the cover image. The distortion of the image results in second drawback, because it is feasible that a steganalytic algorithm can defeat the image steganography and thus reveal that a hidden message is conveyed in a stego image.

The paper will proposes a good approach for steganography using reversible texture synthesis based on edge adaptive and tree based parity check to improve the embedding capacity. A texture synthesis process is of creating a big digital image with a similar local appearance of the original image and has an arbitrary size. And the paper is also using another two methods named edge adaptive and tree based parity check to improve the embedding capacity. The paper fabricates the texture synthesis process into steganography concealing secret messages as well as the source texture. In particular, in contrast to using an existing cover image to hide messages, our algorithm conceals the source texture image and embeds the secret messages through the process of texture synthesis. This allows us to extract the secret messages and the source texture from a stego synthetic texture.

The proposed approach offers three advantages. First, since the texture synthesis can synthesize an arbitrary size of texture images. Since the Human Visual System (HVS) is less sensitive to changes in sharp regions compared to smooth regions, edge adaptive methods has been proposed to find the edge regions and hence improve the quality of the stego image as well as improve the embedding capacity and TBPC to hide the secret data into the cover image. Secondly, a steganalytic algorithm is not to defeat the steganographic approach since the texture image is composed of a source texture rather than by changing the existing image contents. Third, the reversible capability used in the project results in the recovery of the source texture so that the same texture can be used for the second round of message redirect. Within the last decade many advances had been made in the field of digital media, and far obstacle has arisen related to steganography for digital media. Steganography a novel system of knowledge hiding strategies. It embeds messages into a host medium with a purpose to conceal secret messages in order not to arouse suspicion through an eavesdropper. A common steganographic software includes covert communications between two events whose existence is unknown to a viable attacker and whose success will depend on detecting the existence of this communication. Traditionally, the host medium utilized in steganography involves meaningful digital media comparable to digital photo, text, audio, video, 3D model, and many others. A large quantity of photograph steganographic algorithms were investigated with the increasing status and use of digital snap shots. Most photograph steganographic algorithms undertake an present picture as a cover medium. The cost of embedding secret messages into this duvet image is the picture distortion encountered within the stego photograph. This results in two drawbacks. First, on the grounds that the dimensions of the cover picture is fixed, the extra secret messages that are embedded allow for more snapshot distortion. For that reason, a compromise must be reached between the embedding capability and the image high-quality which outcome within the restricted capability supplied in any designated duvet snapshot. Keep in mind that image steganalysis is an strategy used to observe secret messages hidden in the stego image. A stego image includes some distortion, and regardless of how minute it's, this will intervene with the typical facets of the quilt image.

This leads to the 2d concern for the reason that it's nonetheless feasible that an photo steganalytic algorithm can defeat the photo steganography and for that reason reveal that a hidden message is being conveyed in a stego picture. In this paper, we propose a novel process for steganography making use of reversible texture synthesis. A texture synthesis method re-samples a small texture image drawn through an artist or captured in a image in an effort to synthesize a new texture picture with a similar local appearance and arbitrary dimension. We weave the texture synthesis process into steganography concealing secret messages as good because the source texture. In specific, unlike making use of an existing duvet photo to cover messages, our algorithm conceals the supply texture image and embeds secret messages through the approach of texture synthesis. This permits us to extract the key messages and the supply texture from a stego synthetic texture. To the first-class of our capabilities, steganography taking abilities of the reversibility has ever been offered inside the literature of texture synthesis. Our approach offers three advantages. First, considering the fact that the feel synthesis can synthesize an arbitrary dimension of texture graphics, the embedding capacity which our scheme presents is proportional to the dimensions of the stego texture snapshot. Secondly, a steganalytic algorithm is just not more likely to defeat this steganographic technique considering the fact that the stego texture photo consists of a source texture as a substitute than through modifying the prevailing photo contents. 0.33, the reversible capacity inherited from our scheme presents performance to recover the supply texture. When you consider that the recovered supply texture is strictly the identical as the long-established source texture, it can be employed to proceed onto the 2nd round of secret messages for steganography if wanted. Experimental results have demonstrated that our proposed algorithm can provide quite a lot of numbers of embedding capacities, produce visually believable texture snap shots, and get well the supply texture. Theoretical

II LITERATURE SURVEY

Texture synthesis has attained a lot of heeds presently in computer vision and computer Graphic .Most of the present work has concentrated on texture synthesis for example, in which a source texture image is re-sampled using either pixel-based or patch-based algorithms to create a new synthesized texture image with similar appearance and variable size. Pixel-based algorithms create the generated image pixel by pixel and use spatial neighborhood comparisons to select the most closely related pixel in a sample texture as the output pixel. Since each output pixel is identified by the already generated pixels, any wrongly generated pixels during the process influence the rest of the result causing propagation of errors.Otori and Kuriyama in [5] and [6] developed the work of integrating data coding using pixel- based texture synthesis. Secret messages to be hided are encoded into colored dotted patterns and they are directly painted on a blank image. A pixel-based algorithm tunic the rest of the pixels by the pixel-based texture synthesis method, thus masking the existence of dotted patterns. To withdraw messages the printout of the stego synthesized texture image is photographed before applying the data- detecting technique. The capacity given by the method of Otori and Kuriyama depends on the number of the dotted patterns. However, their method has a small error rate of the message extraction.

Patch-based algorithms in [7] and [8] attach patches from a source texture rather than a pixel to synthesize textures. The method of Cohen et al. and Xu et al. improves the image quality of pixel-based synthetic textures because texture structures inside the patches are preserved. However, since patches are attached with a small overlapped region during the synthetic process, one has to make an effort to ensure that the patches agree with their neighbors.

Liang et al. [9] established the patch-based sampling method and used the rowing approach for the coincide areas of nearby patches. Efros and Freeman [10] present a patch stitching methodology called “image quilting.” For each rising patch to be combined and seamed, the algorithm first looks for the source texture and select one candidate patch that fulfill the pre-defined error endurance with respect to neighbors along the overlapped region. Next, a dynamic programming technique is used to confide the minimum error path through the overlapped region. This canonizes an optimal boundary between the selected candidate patch and the synthesized patch, producing visually credible patch stitching.

Ni et al. [11] put forward an image reversible data hiding algorithm which can extract the cover image without any deformation from the stego image after the hidden data have been extracted. Histogram shifting is better technique among existing approaches of reversible image data hiding because it can control the modification to pixels, thus limiting the embedding distortion, and it only requires a small size.

Li et. al. [12] introduced a data hiding method called TBPC to enhance the embedding efficiency by decreasing the unsimilarity between the cover and the stego images. In order to reduce the distortion in the cover pixels, TBPC correspond to the LSB of the cover pixels using a complete N−ary tree. The method in [13] can be developed as another specific matrix embedding, which was improved by Hou et. al. [14], where they put forward a majority-vote parity check (MPC) instead of the original matrix embedding. In [15] Liu et. al. introduced an adaptive steganography algorithm based on block complexity and matrix embedding.

This paper will be taking the advantage of the patch-based method to embed a secret message during the synthesizing procedure. This allows the source texture to be extracted in a message extracting procedure, giving the purpose of reversibility.

III. RELATED WORK

Texture synthesis has bought a lot of awareness not too long ago in pc imaginative and prescient and pc graphics. The most contemporary work has desirous about texture synthesis with the aid of example, wherein a source texture picture is re-sampled utilizing both pixel-established or patch-situated algorithms to produce a brand new synthesized texture photograph with equivalent nearby look and arbitrary dimension. Pixel-based algorithms generate the synthesized snapshot pixel by using pixel and use spatial regional comparisons to pick essentially the most equivalent pixel in a sample texture as the output pixel. For the reason that each output pixel relies on the already synthesized pixels, any wrongly synthesized pixels in the course of the process have an effect on the relaxation of the influence inflicting propagation of blunders. Otori and Kuriyama pioneered the work of mixing data coding with pixel-founded texture synthesis. Secret messages to be hid are encoded into coloured dotted patterns and they're straight painted on a clean snapshot. A pixel-established algorithm coats the rest of the pixels utilizing the pixel-established texture synthesis approach, thus camouflaging the existence of dotted patterns. To extract messages the printout of the stego synthesized texture picture is photographed before applying the information-detecting mechanism. The ability supplied through the procedure of Otori and Kuriyama is determined by the number of the dotted patterns. Nevertheless, their method had a small error rate of the message extraction. Patch-situated algorithms paste patches from a source texture as a substitute of a pixel to synthesize textures. This approach of Cohen et al. And Xu et al. Improves the snapshot great of pixel-established synthetic textures when you consider that texture constructions throughout the patches are maintained. However, due to the fact patches are pasted with a small overlapped region for the duration of the factitious process, one wishes to make an effort to make certain that the patches believe their neighbors. Liang et al. Introduced the patch-centered sampling procedure and used the feathering procedure for the overlapped areas of adjacent patches. Efros and Freeman reward a patch stitching method known as "image quilting". For each new patch to be synthesized and stitched, the algorithm first searches the supply texture and chooses one candidate patch that satisfies the pre-defined error tolerance with respect to neighbors along the overlapped region. Subsequent, a dynamic programming technique is adopted to reveal the minimum error direction by way of the overlapped area. This broadcasts an most advantageous boundary between the chosen candidate patch and the synthesized patch, producing visually plausible patch stitching.

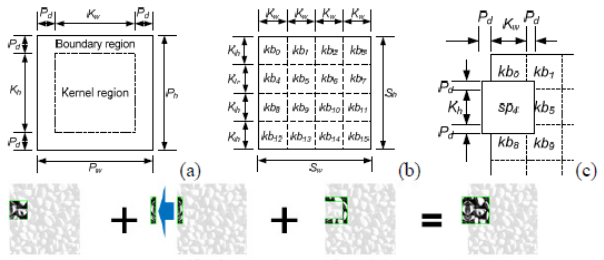

Fig. 1. Patch, kernel blocks, and source patch. (a) The diagram of a patch. The central part of a patch is the kernel region; the other part around the kernel region is the boundary region. (b) An illustration of non-overlapped kernel blocks subdivided from the source texture. (c) The diagram of source patches derived by the expanding process using kernel blocks. (d) The boundary mirroring and expanding for a source patch.

Ni et al. proposed an photo reversible knowledge hiding algorithm which can recuperate the quilt photograph without any distortion from the stego snapshot after the hidden information were extracted. Histogram transferring is a favored manner amongst existing strategies of reversible snapshot knowledge hiding in view that it will probably control the amendment to pixels, consequently limiting the embedding distortion, and it simplest requires a small size location map, thereby decreasing the overhead encountered. The current trendy for reversible photo information hiding is the overall framework provided by way of Li et al. To the quality of our skills, we were unable to reveal any literature that associated patch-headquartered texture synthesis with steganography. On this paper, we present our work which takes competencies of the patch-headquartered methods to embed a secret message during the synthesizing procedure. This allows the supply texture to be recovered in a message extracting approach, delivering the functionality of reversibility. We detail our approach within the next section.

IV. PROPOSED WORK

We illustrate our proposed procedure in this section. First, we will define some normal terminology for use in our algorithm. The basic unit used for our steganographic texture synthesis is known as a "patch." A patch represents an picture block of a source texture the place its dimension is consumer-specific. Fig. 1(a) illustrates a diagram of a patch. We are able to denote the size of a patch by using its width (Pw) and height (Ph). A patch includes the principal part and an outer phase where the primary part is referred to because the kernel vicinity with measurement of Kw×Kh, and the section surrounding the kernel region is known as the boundary region with the depth (Pd). Subsequent, we describe the suggestion of the kernel block. Given a source texture with the size of Sw×Sh we will subdivide the supply texture into a quantity of non-overlapped kernel blocks, each of which has the size of Kw×Kh, as shown as Fig. 1(b). Let KB characterize the gathering of all kernel blocks thereforeKBrepresent the number of factors on this set. We will rent the indexing for each supply patch kbi, i.E., KB=kbi0KB. As an example, given a source texture with the dimension of Sw×Sh =128×128, if we set the scale Kw×Kh as 32×32, then we are able tosixteen kernel blocks. Every aspect in KB (a) (b) (c) 3 can be identified as {kb0, kb1, …, kb15}. We can expand a kernel block with the depth Pd at each side to produce a source patch. The expanding process will overlap its neighbor block. Fig. 1(c) indicates the boundary region of source patch sp4 when we expand the kernel block kb4 to overlap the kernel blocks kb0, kb1, kb5, kb8, and kb9. If a kernel block is located around the boundary of a source texture, we operate the boundary mirroring using the kernel block's symmetric contents to produce the boundary region, as shown in Fig. 1(d) for the kernel block kb4. Similar to the kernel block, we can denote SP as the collection of all source patches and SPn=||SP|| as the number of elements in the set SP. We can employ the indexing for each source patch spi, i.e., SP={spi| i = 0 to ||SP||-1}.

Given a source texture with the size of Sw×Sh, we can derive the number of source patches SPn using (1) if a kernel block has the size of Kw×Kh. In our paper, we assume the size of the source texture is a factor of the size of the kernel block to ease the complexity.

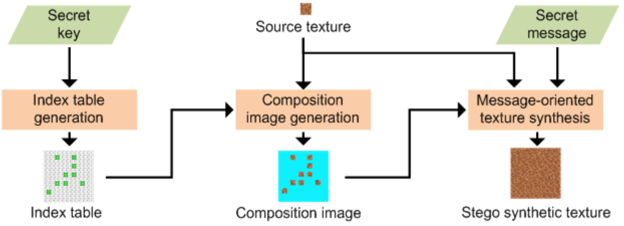

Fig 2. Flowchart of the three-process message embedding procedure

Our steganographic texture synthesis algorithm wishes to generate candidate patches when synthesizing synthetic texture. The proposal of a candidate patch is trivial: we hire a window Pw×Ph and then travel the source texture (Sw×Sh) via transferring a pixel each time following the scan-line order. Let CP=cpizero, 1, …, CPn-1 signify the set of the candidate patches wherequantity of elements in CP. We can derive CPn utilizing (2). When producing a candidate patch, we must make sure that each and every candidate patch is precise; otherwise, we may just extract an fallacious secret message. In our implementation, we employ a flag mechanism. We first verify whether or not the normal supply texture has any replica candidate patches. For a duplicate candidate patch, we set the flag on for the primary one. For the rest of the replica candidate patches we set the flag off to make certain the individuality of the candidate patch within the candidate record. A. Message Embedding approach on this part we will illustrate the message embedding process. Fig. 2 indicates the three strategies of our message embedding method. We will element every procedure in the The primary process is the index desk new release the place we produce an index desk to report the place of the supply patch set SP in the synthetic texture. The index desk allows us to entry the factitious texture and retrieve the supply texture fully. This sort of reversible embedding kind exhibits one of the vital foremost advantages our proposed algorithm offers. We first examine the scale of the index desk (Tpw×Tph). Given the parameters T_w and T_h , which are the width and the height of the artificial texture we intend to synthesize, the quantity of entries on this index table can be decided using (3) the place TP_n denotes the quantity of patches in the stego artificial texture. For simplicity, we chose proper parameters for T_w, T_h, P_w, P_h, and P_d, in order that the quantity of entries is an integer. As an instance, if $T_w \times T_h = 488 \times 488$, $P_w \times P_h = 48 \times 48$, and Pd=8, then we are able to generate an index desk (12×12) containing a hundred and forty four entries. After we distribute source texture to gain the way of reversibility, the source patches will also be allotted in a as an alternative sparse method if the bogus texture has a resolution that is a lot higher than that of the supply texture, as shown in Fig. 3(a).

On the opposite, the source patches is also disbursed in a instead dense method if the artificial texture has a decision that is quite greater than that of the source texture, as proven in Fig. 3(b). For the patch distribution, we avoid positioning a source texture patch on the borders of the factitious texture. This will inspire the borders to be produced by using message-oriented texture synthesis, improving the snapshot pleasant of the artificial texture. We additional outline the primary-priority function L1 and the second-priority role L2, for two varieties of precedence locations the place and 4), characterize the quantity in the first-precedence and 2d-precedence positions, respectively.

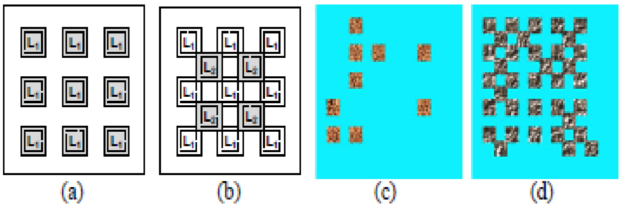

Fig. 3. An illustration of two types of priority locations to paste the source patches: (a) the first-priority locations L1 and (b) the second-priority locations L2; (c) 9 source patches using L1 resulting in sparse distribution; (d) 36 source patches using L1 first, and then employing the L2 leading to the dense distribution.

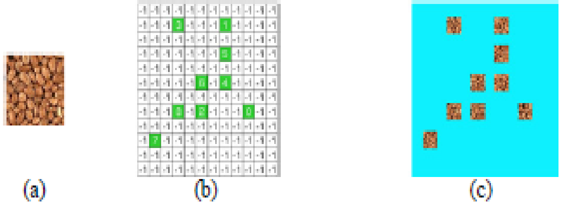

Fig. 4. An illustration of composition image; (a) the source texture (96×96), (b) the index table after patch distribution, (c) the composition image (488×488) by referring (a) and (b).

2) Patch Composition Process

The 2d approach of our algorithm is to paste the source patches right into a workbench to provide a composition photograph. First, we establish a blank image as our workbench where the dimensions of the workbench is equal to the factitious texture. By using referring to the supply patch IDs saved in the index table, we then paste the source patches into the workbench. During the pasting method, if no overlapping of the supply patches is encountered, we paste the supply patches immediately into the workbench, as shown in Fig. Three(c). However, if pasting areas intent the supply patches to overlap every other, we hire the snapshot quilting method to slash the visible artifact on the overlapped area.

Message-oriented Texture Synthesis Process

We have now now generated an index table and a composition photo, and have pasted source patches instantly into the workbench. We will be able to embed our secret message via the message-oriented texture synthesis to produce the final stego synthetic texture. The three primary variations between our proposed message-oriented texture synthesis and the conventional patch-headquartered texture synthesis are described in table I. The first change is the form of the overlapped area. For the duration of the traditional synthesis method, an L-shape overlapped field is in most cases used to assess the similarity of each candidate patch. In distinction, the form of the overlapped field in our algorithm varies on account that we now have pasted source patches into the workbench. Thus, our algorithm wishes to provide more flexibility in order to cope with a quantity of variable shapes shaped via the overlapped area. The 2nd difference lies in the approach of candidate resolution. In traditional texture synthesis, a threshold rank is most commonly given in order that the patch may also be randomly selected from candidate patches when their ranks are smaller than the given threshold.

In distinction, our algorithm selects “suitable” patches by using deliberating secret messages. Finally, the output of the conventional texture synthesis is a pure synthetic texture. However, our algorithm produces a so much one of a kind synthetic texture. The source texture being modified into a quantity of source patches has been pasted as part of the contents within the tremendous synthetic texture. Additionally, the output tremendous texture has been concealed with the secret message. While the conventional texture synthesis algorithm has an “L-shape” overlapped discipline, our algorithm may just accumulate one more four shapes of the overlapped area, as shown in Fig. 5. Count on that the feel synthesis is carried on utilising the scan-line order. The texture area displays a ordinary “L-shape” of an overlapped area, as proven in Fig. 5(a). However, when a local pasted supply patch has occupied the proper part of the working place, this leads to a “downward U-form” of the overlapped area (Fig. 5(b)). In addition, if a regional pasted supply patch has occupied the backside aspect, this results in a “rightward U-shape” of the overlapped field (Fig. 5(c)). If a local pasted source patch has occupied the minimize correct corner of the working area, this leads to a disjointed overlapped subject containing an “L-shape” and a small but isolated section (Fig. 5(d)).

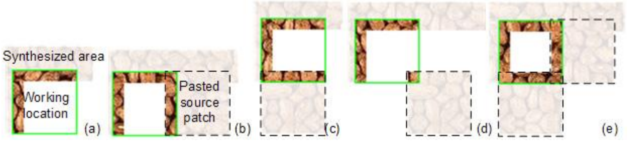

Fig. 5. Five different shapes of the overlapped area may occur during our message-oriented texture synthesis algorithm. Clear textures inside the green square frame represent textures that have been synthesized or textures that are overlapped with the pasted source patches. The blank area inside the green square frame represents the working location which is going to be synthesized

Capacity Determination

The embedding ability is one difficulty of the information embedding scheme. Table II summarizes the equations we described to analyze the embedding potential our algorithm can present. The embedding potential our algorithm can present is said to the ability in bits that may be concealed at each and every patch (BPP, bit per patch), and to the quantity of embeddable patches within the stego synthetic texture (EPn). Each patch can conceal as a minimum one bit of the key message; consequently, the curb certain of BPP will be 1, and the the maximal potential in bits that may be concealed at each and every patch is the upper sure of BPP, as denoted by means of BPPmax. In distinction, if we will decide on any rank from the candidate record, the higher sure of BPP will probably . The complete ability (TC) our algorithm can offer is proven in (5) which is the multiplciation of BPP and EPn. The number of the embeddable patches is the difference between the quantity of patches in the synthetic texture (TPn) and the number of supply patches subdivided in the source texture (SPn).

C. Source Texture Recovery, Message Extraction, and Message Authentication Procedure

The message extracting for the receiver facet entails producing the index desk, retrieving the supply texture, performing the texture synthesis, and extracting and authenticating the secret message hid in the stego synthetic texture. The extracting process comprises 4 steps. Given the key key held in the receiver aspect, the same index table because the embedding method will also be generated. The next step is the supply texture recuperation. Each kernel vicinity with the dimension of Kw×Kh and its corresponding order with admire to the measurement of Sw×Sh source texture may also be retrieved by relating to the index table with the size Tpw×Tph. We can then arrange kernel blocks based on their order, thus retrieving the recovered source texture with a purpose to be exactly the same because the source texture. In the 1/3 step, we practice the composition image iteration to stick the source patches into a workbench to produce a composition snapshot by means of regarding the index table.

This generates a composition snapshot that is equal to the one produced within the embedding method. The ultimate step is the message extraction and authentication step, which comprises three sub-steps. The first sub-step constructs a candidate record based on the overlapped area by using referring to the present working region. This sub-step is thesame because the embedding approach, producing the equal quantity of candidate lists and their corresponding ranks. The 2d sub-step is the suit-authentication step. Given the present working region Cur(WL) on the workbench, we discuss with the corresponding stego artificial texture on the identical working region Stg(WL) to check the stego kernel region SKw×SKh. Then, founded on this stego kernel vicinity, we search the candidate record to determine if there is a patch in the candidate list where its kernel region is the equal as this stego kernel region. If this patch is on hand, we consult with it because the matched patch, and denote it as MKw×MKh. Clearly, we can locate the rank R of the matched patch, and this rank represents the decimal value of the secret bits we conveyed in the stego patch when operating the texture synthesis within the message embedding approach. However, if we can not reveal any matched patch in the candidate record the place the kernel region is the identical as the stego kernel area, it implies that the stego kernel region has been tampered with, main to a failure of the message authentication. In this means, we can authenticate and extract all of the secret messages which can be hid in the stego synthetic texture patch by means of patch.

Our process is resistant in opposition to malicious assaults so long as the contents of the stego photograph usually are not changed. With some facet knowledge, for example, our scheme can live to tell the tale the assaults of the picture mirroring or picture rotation through ninety, one hundred eighty, or 270 degrees. Nonetheless, if malicious attacks lead to alteration of the contents of the stego texture image, the message authentication step will justify the authenticity of the secret messages.

IV METHODOLOGY

The proposed method is described as follows. The basic unit of the steganographic texture synthesis is introduced to as a “patch.” A patch represents an image block of a source 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 ExtractionandMessageAuthenticationProcedure*3) *CapacityDetermination*

1. *Concepts involved in Message Embedding Procedure*

The message embedding procedure involves mainly four steps. They are A) Index Table Generation B.Patch Composition Process C.Combined TBPC and Edge Adaptive Process D.Message Oriented Texture Synthesis Generation.

A. Index Table Generation

The first process of this project is the index table generation where here will create an index table to preserve 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.Combined TBPC and Edge Adaptive Process

Embedding capacity is one of the most mportant Requirements for steganography methods, and it is important for steganography process not to leave any noticeable traceable to the human eyes after hiding the secret data. Here will give a hybrid image steganography method that combines edge adaptive and TBPC methods together. The proposed method exploits the high contrast regions of an image as embedding locations. It is known that human eyes cannot discover modifications in the edge areas since they can do in smooth areas. Therefore, the number of hidden bits is on the basis of the variation value between the two pixels of each block. The integration of TBPC leads to a better embedding capacity. Thus, the proposed method mixes up the strengths of edge adaptive and TBPC.

D.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 stego synthetic texture.

2. *Concept Involved In 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.

$$TC = \text{BPP} \times EPn = \text{BPP} \times (TPn - SPn)$$

IV CONCLUSION

This project proposes a reversible steganographic algorithm using texture synthesis based on edge adaptive and tree based parity check. Given an original source texture, first we have to produce a large stego synthetic texture hiding the secret messages. By using a conventional patch-based method the textures are synthesized. The project will also provides reversibility to retrieve the original source texture from the stego synthetic textures, making possible a second round of texture synthesis if needed. This paper also introduce another image steganography method that combines the edge adaptive and TBPC algorithms to heighten the payload and imperceptibility of the stego image, and thus minimizing the possible distortion during the embedding process to minimize the probability of discovering the secret message data from unauthorized users and also resulting in high embedding capacity. Given an usual source texture, our scheme can produce a large stego synthetic texture concealing secret messages. To the great of our skills, we are the first that may exquisitely weave the steganography right into a conventional patch-based texture synthesis. Our method is novel and provides reversibility to retrieve the long-established supply texture from the stego synthetic textures, making viable a 2d circular of texture synthesis if needed. With the 2 methods we've presented, our algorithm can produce visually believable stego synthetic textures despite the fact that the key messages which includes bit “zero” or “1” have an uneven appearance of probabilities. The awarded algorithm is cozy and strong in opposition to an RS steganalysis attack. We consider our proposed scheme presents mammoth advantages and provides an opportunity to prolong steganographic applications.

One feasible future be trained is to broaden our scheme to support different types of texture synthesis approaches to fortify the snapshot fine of the substitute textures. A further feasible be taught could be to mix other steganography methods to broaden the embedding capacities.

REFERENCE

[1] 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.

[2] 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.

[3] 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.

[4] C. Han, E. Risser, R. Ramamoorthi, and E. Grinspun, “Multiscale texture synthesis,” *ACM Trans. Graph.*, vol. 27, no. 3, 2008, Art. ID 51.

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

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] Li, R.Y., Au, O.C., Lai, K.K., Yuk, C.K., Lam, and S.Y.*: “Data hiding with tree based parity check”.* In: 2007 IEEE International Conference on Multimedia and Expo,pp. 635–638. IEEE (2007)

[13] Hou, C.L., Lu, C., Tsai, S.C., Tzeng, W.G.*: “An optimal data hiding scheme with tree-based parity check.*” IEEE Transactions on Image Processing 20(3), 880–886 (2011)

[14] Liu, G., Liu, W., Dai, Y., Lian, S.: *Adaptive steganography based on block complexity and matrix embedding .Multimedia Systems*20 (2), 227–238(2014)

[15] X. Li, B. Li, B. Yang, and T. Zeng, *“General framework to histogram shifting-based reversible datahiding,IEEETrans.ImageProcess”*vol.22,no.6,pp.2181–2191,Jun.2013.