Steganographic Data Embedding with Revocable Texture Synthesis

Akshay U.Joshi1, Sachin K S2, Vishal .M3, Aruna Kumara B4
1, 2, 3 UG Scholar, Dept. of CSE, RRCE, Bengaluru, 4Assistant Professor, Dept. of CSE, RRCE, Bengaluru
arun.bethur87@gmail.com

ABSTRACT — we propose a unique approach for steganography using a reversible texture synthesis. A texture synthesis method re-samples a smaller texture image that synthesizes a brand new texture image with the same native look and impulsive size. We weave the feel synthesis method into steganography to conceal secret messages. In distinction to exploitation associate existing cowl image to cover messages, our rule conceals the supply texture image and embeds secret messages through the method of texture synthesis. This permits us to extract secret messages and also the supply texture from a stego artificial texture. Our approach offers 3 distinct benefits. First, our theme offers the embedding capacity that's proportional to the scale of the stego texture image. Second, a steganalytic rule isn't doubtless to defeat our steganographic approach. Third, the reversible capability inherited from our theme provides practicality that permits recovery of the supply texture. Experimental results have verified that our planned rule will give varied numbers of embedding capacities, manufacture a visually plausible texture pictures, and recover the supply texture.

Index Terms—Data embedding technique, example-based method, revocable, steganography, texture synthesis.

I. INTRODUCTION

In the last decade several advances are created within the space of digital media, and far concern has arisen relating to steganography for digital media. Steganography a singular method of knowledge concealment techniques. It embeds messages into a bunch medium so as to hide secret messages therefore as not to arouse suspicion by a snooper. A typical steganographic application includes covert communications between 2 parties whose existence is unknown to a potential attacker and whose success depends on police work the existence of this communication. In general, the host medium employed in steganography includes meaningful digital media like digital image, text, audio, video, 3D model, etc. A large number of image steganographic algorithms are investigated with the increasing quality and use of digital images.

Most image steganographic algorithms adopt an existing image as a canopy medium. The expense of embedding secret messages into this cowl image is that the image distortion encountered within the stego image. This ends up in 2 drawbacks. First, since the dimensions of the quilt image is fastened, the additional secret messages that are embedded give additional image distortion. Consequently, a compromise should be reached between the embedding capability and also the image quality which ends within the limited capability provided in any specific cowl image. Recall that image steganalysis is associate approach accustomed discover secret messages hidden within the stego image. A stego image contains some distortion, and no matter however minute it is, this will interfere with the natural options of the quilt image. This leads to the second disadvantage as a result of its still potential that a picture steganalytic formula will defeat the image steganography and thus reveal that a hidden message is being sent in a very stego image.

In this paper, we tend to propose a unique approach for steganography using reversible texture synthesis. A texture synthesis method re-samples a small texture image drawn by a creative person or captured in a photograph so as to synthesize a brand new texture image with a similar native look and arbitrary size. We tend to weave the texture synthesis method into steganography concealing secret messages furthermore because the supply texture. Especially, in distinction to victimization an existing cowl image to cover messages, our algorithm conceals the supply texture image and embeds secret messages through the method of texture synthesis. This permits us to extract the key messages and also the supply texture from a stego artificial texture. To the most effective of our data, steganography taking advantage of the changeability has ever been conferred at intervals the literature of texture synthesis.

II. RELATED WORKS

Texture synthesis has received plenty of attention recently in computer vision and special effects. The foremost recent work has targeted on texture synthesis by example, within which a supply texture image is re-sampled using either pixel-based or patch-based algorithms to supply a replacement synthesized texture image with similar native look and arbitrary size.

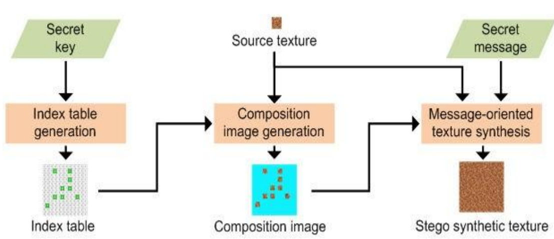
Pixel-based algorithms generate the synthesized image pixel by pixel and use spacial neighborhood comparisons to decide on the foremost similar pixel during a sample texture because the output pixel. Since every output pixel is set by the already synthesized pixels, any incorrectly synthesized pixels throughout the method influence the remainder of the result inflicting propagation of errors.

Patch-based algorithms take patches from a supply texture rather than a component to synthesize textures. This technique of Cohen et al. and Xu et al. improves the image quality of pixel-based artificial textures because the texture structures within the patches are maintained. However, since patches are affixed with alittle overlapped region throughout the artificial method, one must make an attempt to make sure that the patches accept as true with their neighbors.

III. SYSTEM ARCHITECTURE

We illustrate our planned technique during this section. First, we will outline some basic word to be utilized in our rule. The basic unit used for our steganographic texture synthesis is referred to as a ―patch.‖ A patch represents a picture block of a source texture wherever its size is user-specified. illustrates a diagram of a patch. We will denote the dimensions of a patch by its breadth (Pw) and height (Ph). A patch contains the central part and an outer part wherever the central part is observed as the kernel region with size of Kw×Kh, and therefore the half surrounding the kernel region is observed because the boundary region with the depth (Pd).

Next, we tend to describe the conception of the kernel block. Given a source texture with the dimensions of Sw×Sh we will subdivide the source texture into variety of non-overlapped kernel blocks, each of that has the dimensions of Kw×Kh, Let KB represent the gathering of all kernel blocks therefore generated, and ||KB|| represent the quantity of components during this set. We can employ the categorization for every supply patch kbi, i.e., KB= *KB*= {*kbi*| *i* =0 to ||*KB*||-1}. As an example, given a supply texture with the size of Sw×Sh =128×128, if we tend to set the dimensions Kw×Kh as 32×32, then we can generate ||KB||=16 kernel blocks.



The flowchart of the three-process message embedding procedure.

Each part in kb are often known {*kb*0, *kb*1…*kb*15}. We can expand a kernel block with the depth pd at all sides to produce a supply patch. The increasing method can overlap its neighbor block. indicates the boundary region of source patch sp4 once we expand the kernel block kb4 to overlap the kernel blocks kb0, kb1, kb5, kb8, and kb9. If a kernel block is found round the boundary of a supply texture, we operate the boundary mirroring using the kernel block‘s symmetric contents to supply the boundary region, as shown in for the kernel block kb4.

Similar to the kernel block, we are able to denote SP because the collection of all supply patches and SPn=||SP|| because the range of elements within the set SP. we are able to use the classification for every source patch spi, i.e., SP=spi to ||SP||-1}. Given a supply texture with the scale of Sw×Sh, we are able to derive the number of supply patches SPn using (1) if a kernel block has the size of Kw×Kh. In our paper, we have a tendency to assume the scale of the supply texture may be an issue of the scale of the kernel block to ease the complexity.

$$SP_n = \frac{S_w}{K_w} \times \frac{S_h}{K_h} \tag{1}$$

Processing Re-write Suggestions Done (Unique Article) Our steganographic texture synthesis formula should generate candidate patches once synthesizing artificial texture. The thought of a candidate patch is trivial: we've got a bent to use a window Pw×Ph then travel the provision texture (Sw×Sh) by shifting apixel on each occasion following the scan-line order. Let CP=i=0, 1, …, CPn-1 represent the set of the candidate patches where CPn=||CP|| denotes the quantity of elements in CP. We can derive CPn victimization (2).

$$CP_n = \|CP\| = (S_w - P_w + 1) \times (S_h - P_h + 1) \tag{2}$$

In our implementation, we use a flag mechanism. We first check whether or not the first supply texture has any duplicate candidate patches.

A. Message Embedding Procedure

In this section we will illustrate the message embedding technique. shows the three processes of our message embedding procedure. We will explain each process in the following sections.

1) Production of the Index Table

The first method is that the index table generation where we produce an index table to record the placement of the supply patch set SP within the artificial texture. The index table permits us to access the artificial texture and retrieve the supply texture completely. Such a reversible embedding style reveals one among the major advantages our planned algorithmic rule offers.

We 1st confirm the size of the index table (Tpw×Tph). Given the parameters Tw and Th , that square measure the breadth and the height of the artificial texture we have a tendency to shall synthesize, the number of entries during this index table is determined using (3) where total parenteral nutrition denotes the amount of patches within the stego synthetic texture. For simplicity, we selected applicable parameters for Tw, Th, Pw, Ph, and Pd, in order that the amount of entries is associate number. As associate example, if Tw×Th=488×488, Pw×Ph=48×48, and Pd=8, then we will generate associate index table (12×12) containing 144 entries.

$$TP_n = T_{pw} \times T_{ph} = \left\lfloor \frac{(T_w-P_w)}{(P_w-P_d)}+1 \right\rfloor \times \left\lfloor \frac{(T_h-P_h)}{(P_h-P_d)}+1 \right\rfloor \qquad (3)$$

When we distribute source texture to attain the style of reversibility, the source patches may be distributed in a rather sparse manner if the artificial texture contains a resolution that's much larger than that of the supply texture, On the contrary, the supply patches is also distributed in a rather dense manner if the artificial texture contains a resolution that is slightly larger than that of the supply texture. For the patch distribution, we avoid positioning a supply texture patch on the borders of the artificial texture. This will encourage the borders to be made by message-oriented texture synthesis, enhancing the image quality of the artificial texture. We additional outline the first-priority position L1 and also the second-priority position L2, for 2 kinds of priority locations where ‖L1‖ and ‖L2‖, dervied in (4), represent the quantity within the first-priority and second-priority positions, respectively.

Given the amount of patches SPn divided from the supply texture, the strategy of patch distribution is to distribute patches perfectly on the first-priority positions before posting patches on the second-priority positions. Supported the resolution of the synthetic texture, we'll have 2 cases: the thin distribution and dense distribution. These are represented below.

When the amount of supply patches is a smaller amount than or equal to the number of the first-priority positions (SPn≤‖L1‖), the patch will be distributed sparsely. within the security issue section we tend to describe some mathematical analyses of our algorithmic program. The analysis shows that the overall variety of patterns that the thin distribution offers is C^{L1}_{SPn}x SP_n!On the contrary, when the number of supply patches is bigger than the first-priority position (SPn>‖L1‖), the patch are going to be distributed densely. A mathematical analysis shows that the overall variety of patterns that the dense distribution offers is C^{L2}_{SPn-L1}x SP_n!

The index table has the initial values of -1 for every entry, which shows that the table is blank. Now, we'd like to re-assign values after we distribute the supply patch ID within the artificial texture. In our implementation, we tend to use a random seed for patch ID distribution that will increase the safety of our steganographic algorithmic program creating it harder for malicious attackers to extract the supply texture. As a result, the index table are going to be scattered with completely different values where we've got 9 supply patches (no. zero to 8) and one hundred thirty five blank locations with the initial price of –-1‖. During this index table, the entries with non-negative values indicate the corresponding source patch ID divided within the supply texture, whereas these entries with the worth of -1 represent that the patch positions will be synthesized by relating the key message within the message-oriented texture synthesis. Taking the higher than condition into thought, we are able to currently use the random seed Rs to disarrange the ID of the supply patches divided within the supply texture. As an example, if there ar 9 supply patches (SPn=9) and the artificial texture is synthesized with a complete variety of 144 patches (TPn=144), we are able to distribute the disarrayed 9 IDs of the supply patches leading to a thin distribution. Secret messages are going to be encoded within the remaining one hundred thirty five blank locations throughout the message-oriented texture synthesis.

2) Patch Composition Procedure

The second method of our formula is to stick the source patches into a worktable to supply a composition image. First, we establish a blank image as our bench where the scale of the worktable is adequate the artificial texture. By bearing on the supply patch IDs hold on within the index table, we then paste the source patches into the bench. Throughout the pasting method, if no overlapping of the supply patches is encountered, we paste the supply patches directly into the bench. However, if pasting locations cause the supply patches to overlap one another, we use the image quilting technique to reduce the visual object on the overlapped space.

3) Message-based Texture Synthesis Process

We have currently generated an index table and a composition image, and have affixed supply patches directly into the workbench. We‘ll insert our secret message via the message-oriented texture synthesis to provide the ultimate stego synthetic texture. The 3 elementary variations between our projected message-oriented texture synthesis and therefore the standard patch-based texture synthesis area unit delineated in Table I. The first difference is that the form of the overlapped space. During the conventional synthesis method, an L-shape overlapped space is normally accustomed confirm the similarity of each candidate patch. In distinction, the form of the overlapped space in our algorithm varies because we've affixed supply patches into the workbench. Consequently, our algorithmic rule must offer a lot of flexibility so as to address variety of variable shapes formed by the overlapped space.

The second distinction lies within the strategy of candidate selection. In standard texture synthesis, a threshold rank is usually given in order that the patch are often arbitrarily designated from candidate patches once their ranks area unit smaller than the given threshold. In distinction, our algorithmic rule selects –appropriate‖ patches by taking into thought secret messages. Finally, the output of the standard texture synthesis may be a pure synthetic texture

While the traditional texture synthesis algorithmic rule has an –L-shape‖ overlapped space, our algorithmic rule might acquire another four shapes of the overlapped space. The texture space reveals a typical –L-shape‖ of an overlapped area, as shown in .However, once a close-by glued source patch has occupied the correct aspect of the operating location, this ends up in a –downward U-shape‖ of the overlapped space.. If a close-by glued supply patch has occupied the lower right corner of the operating location, this leads to a disjointed overlapped space containing AN –L-shape‖ and a tiny low however isolated half .Finally, if 2 near pasted supply patches have occupied the correct and bottom aspect of the operating location, this can contribute to AN –O-shape‖ of the overlapped space .

For each candidate patch inside the candidate list, one of the five shapes of overlapped space represented above can occur once referring to the synthesized space within the operating location. Thus, we can cipher the mean sq. error (MSE) of the overlapped region between the synthesized space and also the candidate patch. After all MSEs of the patches within the candidate list square measure determined. During this method, a section of the n-bit secret message has been concealed into the chosen patch to be affixed into the operating location.

B. Capacity Establishment

The embedding capability is one concern of the data embedding theme. Table II summarizes the equations we tend to described to investigate the embedding capability our algorithmic rule will offer. The embedding capability our algorithmic rule can give is related to the capability in bits that may be concealed at every patch (BPP, bit per patch), and to the quantity of embeddable patches within the stego artificial texture (EPn). Everypatch will conceal a minimum of one little bit of the key message; therefore, the edge of BPP can be 1, and also the the top capability in bits that may be concealed at every patch is that the bound of BPP, as denoted by BPPmax. In distinction, if we can choose any rank from the candidate list, the upper bound of BPP are [$\log_2$(CPn)]. The overall capability (TC) our algorithmic rule can give is shown in (5) that is that the multiplication of BPP and EPn. The quantity of the embeddable patches is that the distinction between the quantity of patches within the synthetic texture (TPn) and also the variety of supply patches subdivided within the supply texture (SPn).

$$TC = \mathrm{BPP} \times EP_n = \mathrm{BPP} \times (TP_n - SP_n)$$

Suppose we offer a supply texture Sw×Sh =128×128, and we will generate an artificial texture Tw×Th=488×488. We specify the patch size Pw×Ph=48×48 and therefore the boundary depth pd =8 pixels. this may cause the range of the BPP between one and 12. we will manufacture SPn=16 source patches and TPn=144 patches on the stego artificial texture. Thus, there square measure EPn =128 embeddable patches. If we take the BPP=12, then the full embedding capability is TC=1536 bit

C. Source/Supply Texture Recovery, Message Extraction and Message Authentication Procedures

The message extracting for the receiver side includes generating the index table, retrieving the supply texture, performing the texture synthesis, and extracting and authenticating the key message concealed within the stego synthetic texture. The extracting technique contains four steps.

Given the secret key command within the receiver side, an equivalent index table as the embedding procedure are often generated. The next step is that the supply texture recovery. Every kernel region with the size of Kw×Kh and its corresponding order with relevance the size of Sw×Sh supply texture are often retrieved by referring to the index table with the size Tpw×Tph. we are able to then organize kernel blocks supported their order, so retrieving the recovered source texture which can be precisely the same because the supply texture. Within the third step, we apply the composition image generation to stick the supply patches into a work table to produce a composition image by referring to the index table. This generates a composition image that's just like the one produced within the embedding procedure. The final step is that the message extraction and authentication step, that contains 3 sub-steps. The primary sub-step constructs a candidate list supported the overlapped space by referring to the present operating location. This sub-step is that the same because the embedding procedure, manufacturing an equivalent variety of candidate lists and their corresponding ranks.

The second sub-step is the match-authentication step. Given the current operating location Cur(WL) on the work table, we refer to the corresponding stego artificial texture at a similar working location Stg(WL) to see the stego kernel region SKw×SKh.

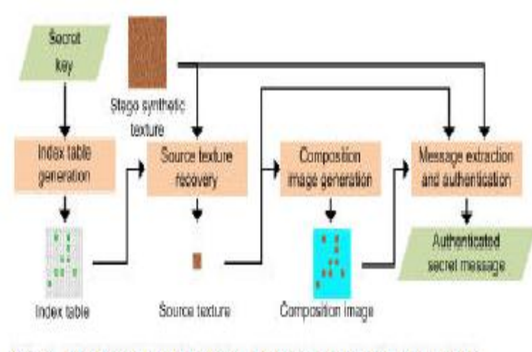


Fig. 7. The flowchart of the four-step message extracting procedure.

Then, based on this stego kernel region, we search the candidate list to see if there's a patch within the candidate list where its kernel region is the same as this stego kernel region. If this patch is out there, we see it as the matched patch, and denote it as MKw×MKh. Clearly, we will find the rank R of the matched patch, and this rank represents the decimal worth of the key bits we sent within the stego patch when operational the feel synthesis within the message embedding procedure. However, if we cannot disclose any matched patch in the candidate list wherever the kernel region is the same because the stego kernel region, it implies that the stego kernel region has been tampered with, resulting in a failure of the message authentication. In this manner, we will authenticate and extract all of the key messages that are hid within the stego artificial texture patch by patch.

Our methodology is resistant against malicious attacks as long as the contents of the stego image don't seem to be modified. With some side information, as an example, our theme will survive the attacks of the image mirroring or image rotation by ninety, 180, or 270 degrees. However, if malicious attacks result in alteration of the contents of the stego texture image, the message authentication step can justify the genuineness of the key messages.

IV. CONCLUSIONS & FUTURE ENHANCEMENTS

This paper proposes a novel revocable steganographic algorithmic rule using texture synthesis. Given an artless supply texture, our scheme will manufacture an outsized stego artificial texture concealing secret messages. To the best of our information, we are the first ones that can fine weave the steganography into a conventional patch-based texture synthesis. Our technique is novel and provides reversibility to retrieve the first supply texture from the stego artificial textures, making doable a second round of texture synthesis if required. With the 2 techniques we have introduced, our algorithmic rule will manufacture visually plausible stego artificial textures even though the key messages consisting of bit ―0‖ or ―1‖ have an uneven appearance of possibilities. The given algorithmic rule is secure and strong against an RS steganalysis attack. We believe our proposed theme offers substantial advantages and provides an opportunity to increase steganographic applications. One potential future study is to expand our theme to support other kinds of texture synthesis approaches to enhance the image quality of the artificial textures. Another potential study would be to mix alternative steganography approaches to increase the embedding capacities.

REFERENCES

1. K. Thangadurai; G. Sudha DeviComputer **―An analysis of LSB based image steganography techniques‖,** Communication and Informatics (ICCCI), 2014 International Conference
2. P. Marwaha; P. Marwaha, **―Visual cryptographic steganography in images‖**, Computing Communication and Networking Technologies (ICCCNT), 2010 International Conference
3. 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.
4. J. L. Rodgers and W. A. Nicewander, **―Thirteen ways to look at the correlation coefficient**,‖ The American Statistician, vol. 42, no. 1, pp. 59-66, 1988.
5. D. Artz,‖**Digital steganography: hiding data within data‖,** IEEE Internet Computing Year: 2001, Volume: 5, Issue: 3 Pages: 75 - 80, DOI: 10.1109/4236.935180
6. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, **―Image quality assessment: from error visibility to structural similarity**,‖ IEEE Trans. Image Process., vol. 13, no. 4, pp. 600-612, 2004.
7. D. Singla; M. Juneja **―An analysis of edge based image steganography techniques in spatial domain‖**Recent Advances inEngineering and Computational Sciences (RAECS)‖, 2014