 CMC, vol.55, no.1, pp.151-163, 2018

Steganography Using Reversible Texture Synthesis Based on Seeded Region Growing and LSB

Qili Zhou[1], Yongbin Qiu[1], Li Li[1, *], Jianfeng Lu [1], Wenqiang Yuan[1], Xiaoqing Feng[2] and Xiaoyang Mao[3]

Abstract: Steganography technology has been widely used in data transmission with secret information. However, the existing steganography has the disadvantages of low hidden information capacity, poor visual effect of cover images, and is hard to guarantee security. To solve these problems, steganography using reversible texture synthesis based on seeded region growing and LSB is proposed. Secret information is embedded in the process of synthesizing texture image from the existing natural texture. Firstly, we refine the visual effect. Abnormality of synthetic texture cannot be fully prevented if no approach of controlling visual effect is applied in the process of generating synthetic texture. We use seeded region growing algorithm to ensure texture's similar local appearance. Secondly, the size and capacity of image can be decreased by introducing the information segmentation, because the capacity of the secret information is proportional to the size of the synthetic texture. Thirdly, enhanced security is also a contribution in this research, because our method does not need to transmit parameters for secret information extraction. LSB is used to embed these parameters in the synthetic texture.

Keywords: Steganography, texture synthesis, LSB, seeded region growing algorithm, information segmentation.

1 Introduction

With the development of media and information technology [Gurusamy and Subramaniam (2017); Pradeep, Mridula and Mohanan (2016)], digital multimedia is popular in network transmission. Users are benefitted but information security is challenged. Information hiding technology has be-come a hot topic in the field of information security. Information hiding has been used in data confidentiality, identity authentication, copyright protection, piracy tracking and so on. Information hiding technology is accompanied with bench-marks such as insensitivity, robustness, and so on. Information hiding is divided into information hiding of image, video, audio, text and so on according to different carriers. Digital images are ideal as information hiding carrier because they are popular and contain a lot of information. Methods include space domain and transform domain depending on

[1] School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, 310018, China.

[2] School of Information, Zhejiang University of Finance & Economics, Hangzhou, 310000, China.

[3] University of Yamanashi, Yamanashi-ken, 409-3866, Japan.

* Corresponding Author: Li Li. Email: lili2008@hdu.edu.cn.

the current image information hiding scheme. Least Significant Bit (LSB) is one of the classical spatial domain algorithms, whereas the representative of the frequency domain algorithms is the spread spectrum algorithm proposed by Cox et al. [Cox, Kilian, Leighton et al. (1996)].

In order to fundamentally resist the detection of all kinds of cryptographic analysis algorithms, the Coverless Image Steganography Without Embedding information hiding has been proposed in recent years. Compared with classical information hiding, coverless images do not need extra carriers except image itself, gets secret information directly from stego carriers. However, it is hard to fast and effectively hide of secret information, because there is no perfect image feature description method to map the original image with secret information.

The generated or synthetic images with digital image processing are used in in-formation hiding besides the classic images. The texture images are resampled based on pixels or texture blocks to generate synthetic texture images with similar local appearance and arbitrary size. Secret information is embedded in the process of synthesis.

We improve steganography using reversible texture synthesis [Wu and Wang (2014)]. Firstly, the secret information is encrypted with RSA Asymmetrical Encryption Algorithm. Secondly, stego synthetic texture is constructed according to the secret information. Thirdly, the seeded region growing method is used as synthetic algorithm of texture blocks to increase the rationality of the texture. Finally, parameters for secret information extraction are embedded by the LSB method in the synthetic texture. The experimental results show that our synthetic texture is natural and has a good display effect.

2 Related work

Improvements to classical steganography methods yield good results. Zheng et al. [Zheng and Chen (2010)] proposed an improved bit-plane information hiding algorithm based on RGB color space. The secret information is embedded in blocks of RGB channels, because human eyes have different visual sensitivity to RGB. Their result has advantages over the direct bit-plane replacement algorithm, and information embedding capacity is higher. The algorithm meets the requirement of information hiding, but it is hard to keep robustness.

Huang et al. [Huang and Yang (2011)] discussed the superiority and insufficiency about Image Information Hiding Technology based on LSB. Secret information is randomly embedded in the lowest level of the 8-bit plane of insignificant pixels. The secret information is made invisible. However, the algorithm is less robust and vulnerable to external attack due to the usage of insignificant pixels.

Niu et al. [Niu and Zhang (2017)] proposed a reversible data hiding algorithm based on image interpolation. The scheme has a high embedding capacity and security. This method is limited by the need for adaptive prediction of texture images, and some image pixels are deleted in the process.

As to coverless image steganography without embedding, Zhou et al. [Zhou, Sun, Harit et al. (2016)] proposed a coverless image steganography framework. It need not employ designated cover image for embedding secret information. Image hash sequence is generated when the sum of pixels between two blocks in the same image is compared. An image is represented by a single hash sequence. A series of images are found according to the secret information. These images are regarded as stego images since they already contain secret data. The disadvantage of this approach is that the hidden data are limited to 8 bits each image.

As to robustness, Zheng et al. [Zheng, Wang, Ling et al. (2017)] proposed a new coverless steganography method based on SIFT features. Image hash sequence is generated with the orientation information of the SIFT feature points. Images containing secret information and an additional image are transmitted to receiver. Some important parameters are carried by the additional image. Based on the same principle, Zhou et al. [Zhou, Cao and Sun (2016)] introduced a coverless information hiding method based on the bag-of-words model (BOW). A set of sub-images related to the text information is found to be sent. These methods need additional information to transmit the necessary parameters and it is a conundrum to have a better algorithm to match images and secret information.

In recent years, texture construction is used to hide information [Qian, Zhou, Zhang et al. (2017)]. Pan et al. [Pan, Qian and Zhang (2016)] proposed a new steganography method using texture construction. The principle is algorithm selects different shapes and background elements according to the secret information. With pre-defined transforming functions, the data-hider transforms the painted image to construct a stego image. The constructed stego image has good visual effects, secret messages are properly hidden in the image. Different volumes of information can be hidden with different shape parameters.

Wu et al. [Wu and Wang (2014)] proposed a novel approach for steganography using a reversible texture synthesis. A texture set is resampling the smaller collection of texture images, and the texture set is made up of similar local appearance of new texture image. Steganography and texture synthesis are combined to hide secret information. Compared with the method of using the cover image to hide information, their algorithm hides the source texture image and embed the secret information through the process of texture synthesis, allows recovery of the source texture during the process of message extraction. Different quantities of embedding capacity are made to produce a visually plausible texture image.

Based on classical steganography, texture synthesis and coverless image steganography, we propose the steganography using reversible texture synthesis based on seeded region growing and LSB. While the secret information is hidden in the stego texture image, the stego texture image still has similar properties to the source texture. By using seeded region growing method to synthesize texture, image remains natural and realistic, has good display effect. We give details in the next sections.

3 Proposed method

The work by Wu et al. [Wu and Wang (2014)] is a path breaking study on steganography. However, we put forward a few problems. Firstly, secret messages are not encrypted and therefore vulnerable. Secondly, texture synthesis capacity is not fully used because the length of the secret message is variable. Thirdly, the visual effect of the texture image is impacted because the similarity among texture patches is only simply evaluated. And what is more, there is risk in the image transmission because not only the synthetic texture, but also some additional parameters need to be transmitted.

The above problems considered, we propose an approach for steganography using a reversible texture synthesis based on seeded region growing and LSB. The synthetic texture is obtained by reassembling the current texture while secret messages are embedded. Security is important, the secret messages are transformed into strings using of image splitting is used in texture synthesis, the visual effect is therefore improved for the synthetic texture. Additional parameters are concealed using LSB based on the characteristics of PNG image, the security in the transmission of the encrypted strings is enhanced.

3.1 Message embedding procedure

The message embedding process is shown in Fig. 1. Details are described in the next five sections.

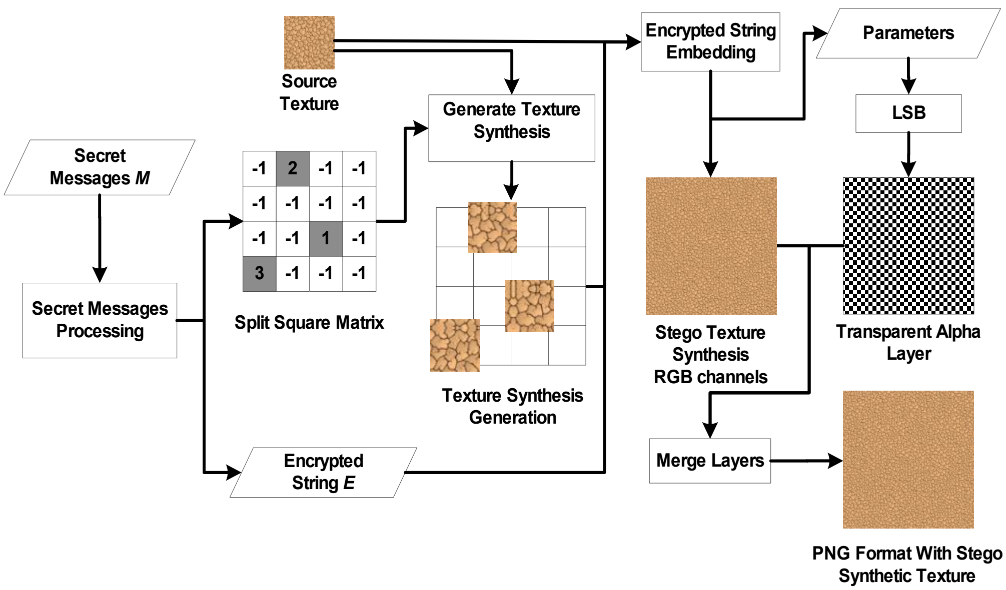


Figure 1: The flowchart of secret messages embedding procedure

A brief introduction to the symbols and parameters in Fig. 2 is as follows. As shown in Fig. 2(a), the source texture is split into kernel blocks ($K_w \times K_h$). KB represents the set of kernel blocks. P_d represents the size of the overlaid boundary. As shown in Fig. 2(b) and Fig. 2(c), these patches ($P_w \times P_h$) are the source patches which is obtained by extending kernel blocks. The set of the patches is represented by SP, the number of elements in the set SP is represented by SP_n which is $\|SP\|$.As shown in Fig. 2(d), the mirror image that has the same size in the kernel block is copied to the ex-tended area if kernel block is located in the source texture's boundary (the cases of kb_1 and kb_5) in the process of extending boundary. w, the step length of alternative blocks, is then determined. CP represents the set alternative blocks ($P_w \times P_h$) and it is obtained according to w in the extended source texture. The number of elements in the set CP is represented by CP_n which is $\|CP\|$.

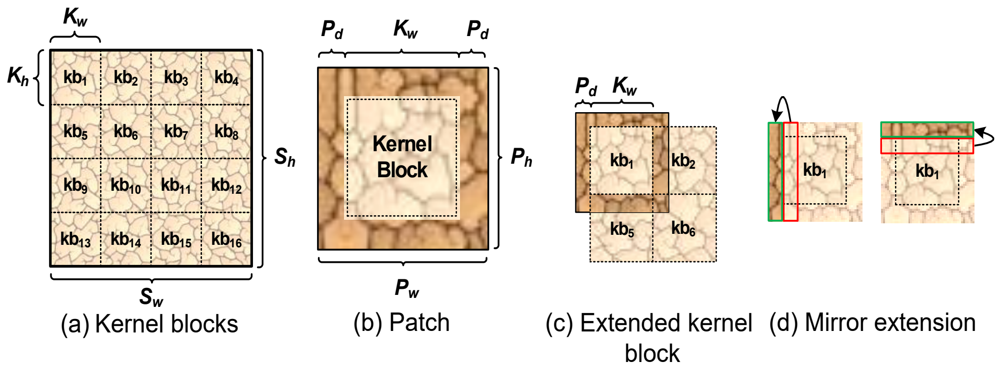


Figure 2: (a) Kernel blocks. (b) Source patch. (c) Extended kernel block and (d) Mirror extension. (a) Segmentation of non-overlapping kernel blocks from source texture; (b) The inner part of the dotted line is the kernel block, and the peripheral area is the extended boundary; (c) The source texture is extended from the kernel block to get the source patches; because the kb_1 is located at the source texture boundary, the left and the upper part needs boundary expansion; (d) The red area mirrors and is copied to the green area

3.2 Secret messages encryption based on asymmetric encryption

The asymmetric encryption, RSA, is selected in the secret messages encryption [Zhang and Liu (2011)]. A pair of keys (public key and private key) is used in the asymmetric encryption. And their names (public and private) explicitly illustrate their functions.

The receiver publishes its public key on the net whereas keeps its private key. The sender uses this public key (on the net by the receiver) to encrypt secret message M (digital string) and encrypted string E (digital string) is obtained. The receiver decrypts E using the private key and the secret message M is recovered. This process is shown in Fig. 3.

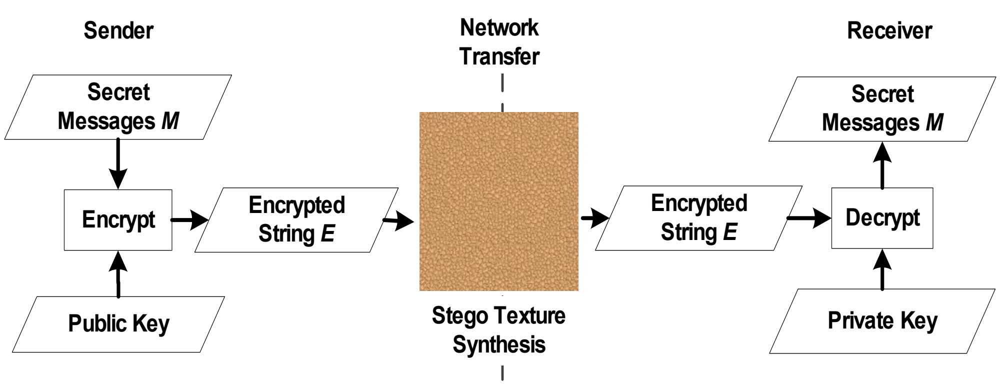


Figure 3: Information transmission via public key and private key

3.3 Encrypted string E segmentation with a square matrix

String E is segmented with a square matrix (a matrix with the same number of rows and columns). The encrypted string E consists of decimal digits. Every digit is a character of the secret message and embedded into synthesized texture, as shown in Fig. 4(a). It results in redundancy of non-embedded blocks, if only one character of E is embedded. The redundancy in turn results in space waste and increase extraction difficulty. E is segmented in such groups, as shown in Fig. 4(b).

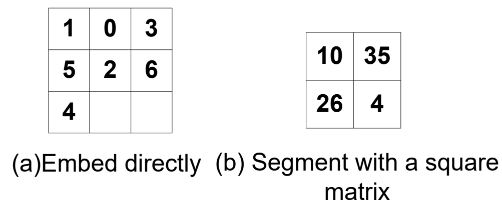


Figure 4: (a) Embed directly and (b) Segment with a square matrix

Assume that M is '2926102129', and SP_n is 2, the process of segmenting E is as follows.

1) The length of E is: len is 10, from Eq. (1) and (2), the size of square matrix is: $T_{pw} \times T_{ph}$ is 3 × 3, the capacity of matrix is: TP_n is 9.

$$T_{pw} = T_{ph} = \lfloor \sqrt{len + SP_n} \rfloor \quad (1)$$

$$TP_n = T_{pw} \times T_{ph} \quad (2)$$

2) From Eq. 3, capacity of stego texture synthesis is: TS_n is 7. Because $TS_n \leq len$, the texture synthesis does not have enough space for every digit.

$$TS_n = TP_n - SP_n \quad (3)$$

3) From Eq. 4, the delta of capacity is: d_t is 7, d_t digits of the secret message need to be merged.

$$d_t = len - TS_n \quad (4)$$

4) Locating suitable digits for the mergence. Find digit '1' and merge '1' with the next digit. The number of digit '1' in M is 2, c_1 is 2. The set of digit '1' merged with the next digit is $Q_1 = \{10,12\}$. The elements in Q_1 are listed in ascending order. Q_1 is referred in sorting if $c_1 \leq d_t$, and updated d_t is $d_t - c_1$. Only the front d_t elements in Q_1 are sorted if $c_1 > d_t$, and updated $d_t \leq 0$.

5) M are further merged as follows. Step 4) is repeated, m_i is merged with the next digit, 'm_i' $\{m_i | i = 2,3,\ldots,9\}$. The process does not stop till $d_t \leq 0$. The result of splitting M is as follows:

'2 | 9 | 26 | 10 | 2 | 12 | 9' ('|' is delimiter)

This mergence effectively decreases the space for embedding data and guarantees the digits are as small as possible after the mergence. The quality of texture synthesis is guaranteed and the impact to the final selecting *Rank* in *CP* blocks.

3.4 Index matrix construction and workbench for texture synthesis

An index square matrix and the workbench for texture synthesis should be constructed before the encrypted string E is embedded into texture synthesis. The index matrix records the location of the texture blocks in texture synthesis. Texture synthesis space is the storage for the texture blocks recorded in the index matrix. Index square matrix $\left(T_{pw} \times T_{ph}\right)$ is obtained in pre-processing secret messages. Through the known overlaid boundary P_d and the size of the source texture block$(S_w \times S_h)$, the synthetic texture size $(T_w \times T_h)$ can be calculated by Eq. (5). And each coordinate in the index matrix is related to a region in the corresponding texture area $(S_w \times S_h)$. An example of the index square matrix and the synthetic texture workbench is shown in Fig. 5.

$$\begin{cases} T_w = P_d + T_{pw} \times (S_w - P_d) \\ T_h = P_d + T_{ph} \times (S_h - P_d) \end{cases} \tag{5}$$

As shown in Fig. 5(a), the grey cells in the index square matrix is numbered p_i $\{p_i | i = 1,2,\ldots,TP_n\}$, the texture block sequence number represents the embedding order. The texture blocks are embedded according to the number in the index matrix, as shown in Fig. 5(b). The regions related with the cells with initialized '-1' are not embedded with the texture. The black border is the overlaid region of texture blocks, the width of the black border is P_d, as shown in Fig. 5(b). An index matrix and the synthetic texture workbench are obtained in this way.

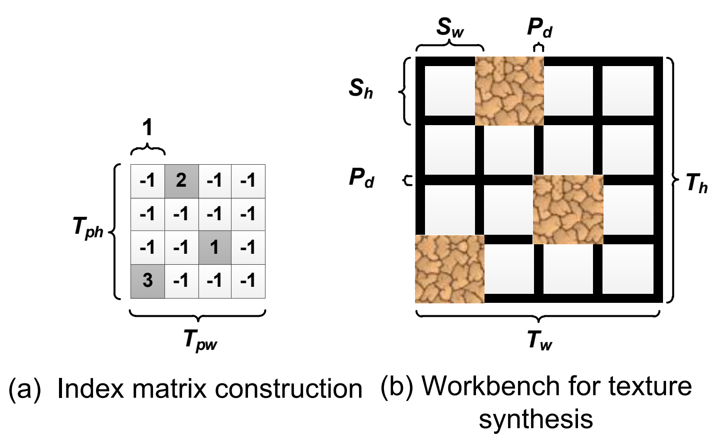


Figure 5: An example of index matrix and synthetic texture workbench. (a) Index matrix construction and (b) Workbench for texture synthesis

3.5 Embedding encrypted string and synthetic texture through seeded region growing algorithm

Seeded region growing algorithm is used as texture synthesis strategy in encrypted string embedding process. We made improvement to texture block embedding [Wu and Wang (2014)]. The seeded region growing is used, the starting points are several randomly distributed seeds (a block with embedded texture) and grow synchronously. The character in-formation that is suitable for encrypted string E is selected. The CP block that is the nearest to the embedded texture is embedded into workbench. This helps the synthetic texture with good visual effect. Steps are as follows.

1) Initialize index matrix with source texture. SP_n blocks are assigned with the coordinate of seed point, the label of every seed point is p_i $\{p_i|i = 1, 2, \ldots, TP_n\}$ and is recorded in index matrix. The labelled source texture blocks are embedded into workbench.
2) Source texture blocks serve as starting points. The texture blocks (with p_i is 1) are the starting point and grow sideward in all four-directions. BFS (Breadth-First Search) is used and four neighbors are optional in the growth process, as shown in Fig. 6. One of the four directions is selected as starting direction. The sequence in our experiment is up-down-left-right. A round of growth consists of growth once in every of these four directions. The new starting point becomes $p_i + 1$ after the previous growth round. Any grown block is skipped if it is met by BFS.

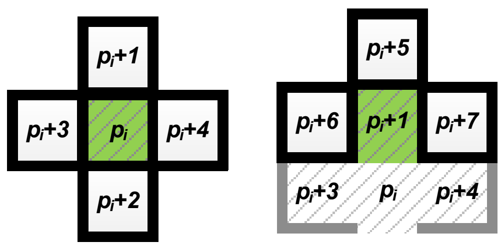


Figure 6: Seeded region growing strategy

3) Embed encrypted string E. Alternative blocks in the set of CP are selected and overlaid with embedded texture. Information is embedded in the overlay process. As shown in Fig. 7, the embedding block CP_i is overlaid with the embedded texture in the black area. MSE in black area of both block CP_i and the embedded texture is calculated. The set of CP is sorted in ascending order of MSE. $Rank$ is assigned according to r_i $\{r_i|i = 0, 1, \ldots, CP_n - 1\}$. The next selected embedding block is r_i according to the contents of encrypted string E. The label of this is p_i $\{p_i|i = SP_n + 1, SP_n + 2, \ldots, TP_n\}$.

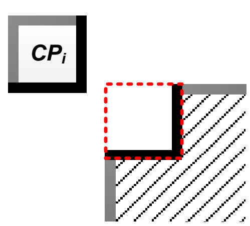


Figure 7: Overlay of embedding texture block with embedded texture

4) Step 2) and 3) are repeated according to the sequence in the index matrix till the embedding in synthetic texture workbench is completed. The stego synthetic texture is obtained.

3.6 PNG transparent layer parameters hiding based on LSB

The size of kernel block ($K_w \times K_h$), the size of source texture ($P_w \times P_h$), and the sliding window size of the selected alternative block w need to be transmitted additionally when the synthetic texture image is transmitted in [Wu and Wang (2014)]. The additional transmission is harmful to security. So, we hide these additional parameters based on LSB. LSB is used to hide parameters in PNG transparent Alpha layer. The visual effect is never impacted because no pixel in the synthetic texture is changed. The steps are as follows.

1) Convert parameters K_w, P_d, w, T_c into binary form. T_c is obtained by Eq. (6). The elements in T_c denote the coordinate $(T_{pw} \times T_{ph})$ of each stego texture of SP.

$$T_c = \{(T_{pi} - 1) \times T_{pw} + T_{pj} | T_{pi} = 1 \text{ to } T_{pw}, T_{pj} = 1 \text{ to } T_{ph}\} \tag{6}$$

2) Obtain the step length of parameter extraction E_w and converting it into binary form. The step length of parameter extraction is the same as the binary parameter string extraction. E_w is the longest among all step lengths. Every parameter is padded with '0' according to E_w.
3) Concatenate parameters in T_c and obtains its length. The elements in the set of T_c is obtained. K_w, P_d, w and T_c form a binary string t_s. Its length len_t is also converted into binary form.
4) A zeros square matrix A_t with dimension $(T_{pw} \times T_{ph})$ is used as PNG transparent layer, as shown in Fig. 8(a).

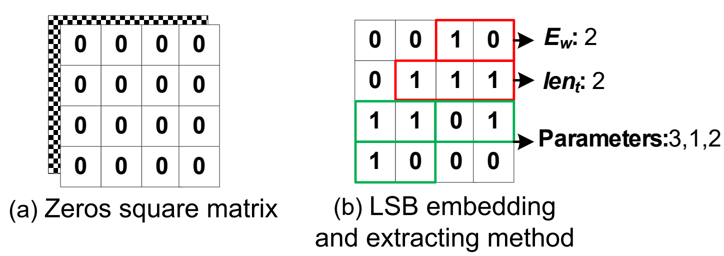


Figure 8: (a) Zeros square matrix and (b) LSB embedding and extracting method

5) Embed parameters. The splitting step length and string length are embedded into the end of the first and second row in A_t. t_s is embedded from the beginning of the third row and onwards. It is easy for extraction to embed the binary string in the end, as shown in Fig. 8(b).
6) Extraction starts when the first '1' is accessed. The process is easy because the length of every string need not to be record.

7) Synthesize PNG texture. The image with four channels is synthesized with stego texture after calculating $255 - A_t$. The PNG image with synthetic texture is obtained.

3.7 Secret messages extraction

The receiver gets a synthetic texture image with no additional information, as shown in Fig. 9. Message extraction mainly consists of five steps: obtaining parameters, generating index tables, recovering source texture, texture seed point embedding and texture synthesis, extracting and verifying E in hidden synthetic texture, and recovering secret messages M through key.

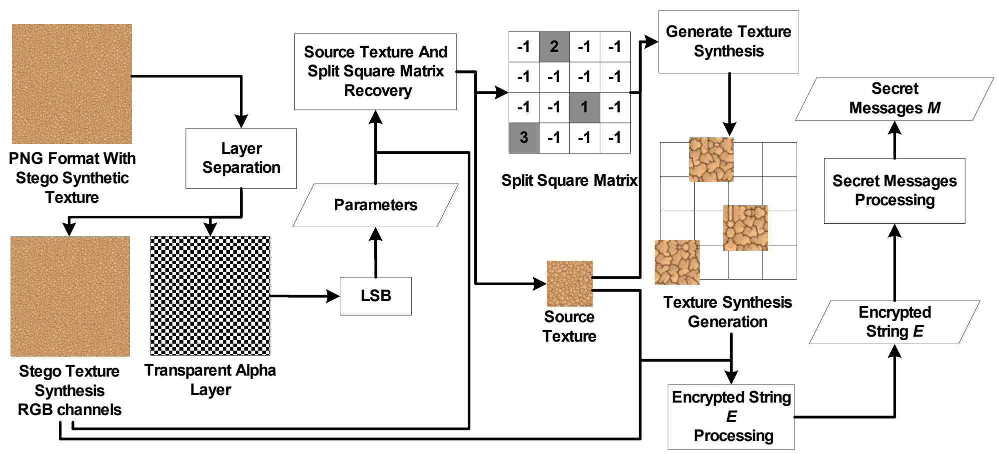


Figure 9: The flowchart of secret messages extracting procedure

The steps of the extraction are as follows:

1) The RGB three channels and the transparent Alpha channel are separated from the synthetic texture, and the additional parameters K_w, P_d, w, t_s are obtained through the LSB extraction method.

2) Source texture and index matrix recovery. The index matrix is reconstructed by the obtained additional parameters. The coordinates of the source texture block $(P_w \times P_h)$ in the matrix is obtained by reversed operation to the T_c set. The corresponding kernel blocks $(K_w \times K_h)$ are extracted from the synthetic texture according to these coordinates. The kernel blocks are arranged in order to obtain a recovered source texture which is exactly as same as the source texture.

3) According to step 1) the parameters extracted and the source texture is rebuilt. The source patches set $\widetilde{CP}$ and candidate blocks set $\widetilde{CP}$ are generated by the new source texture. $\widetilde{SP}$ is embedded into the new workbench. The recovered synthetic texture is generated by seeded region growing. In the process of synthesis, $\widetilde{CP}$ is arranged in ascending order, so as to get the same *Rank*.

4) Extract the encrypted string E. Unlike the embedding candidate blocks, the kernel block $(K_w \times K_h)$ is extracted from the synthetic texture block $(P_w \times P_h)$ at the same coordinates. The extracted kernel block is checked whether it is the same as any

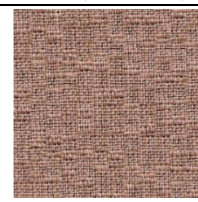
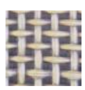
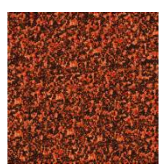
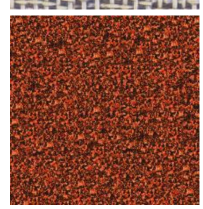
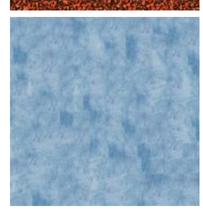
kernel block of the candidate blocks in the *CP*. If the same block is found, the corresponding *Rank* of the candidate block is the character of the encrypted string *E*. Otherwise, the embedded block is tampered or damaged and the encrypted string *E* is not valid.

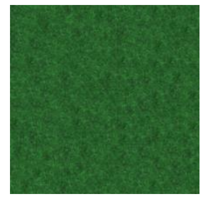
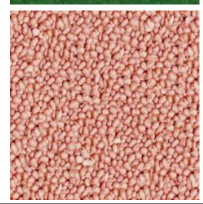
5) The receiver decrypts the acquired encrypted string *E* with the private key. The original secret information *M is obtained*.

4 Experimental results

Our proposed algorithm is compared to the previous work. The secret information of either 50-bit or 150-bit decimal is embedded in the 128×128 source texture. The kernel blocks of 32×32 are selected, and the extended boundary P_d is 8. The stego synthetic textures are generated according to the embedding algorithm in Section 3. The synthetic textures are shown in Tab. 1.

Table 1: Results of stego synthetic textures

Name	Source Textures (128×128)	Synthetic Textures 1 (328×328)	Synthetic Textures 2 (488×488)
Cane 1			
Cane 2			
Flame			
Ice 1			

Ice 2			
Lawn			
Peanuts			

Compared with Wu et al. [Wu and Wang (2014)], we have made improvement in the aspects in Tab. 2.

Table 2: A fundamental of five differences between method in Wu et al. [Wu and Wang (2014)] and ours

	[Wu and Wang (2014)]	Ours work
Shapes of the overlapped area	Five different shapes in all	Any shapes generated in the direction of seeded region growing
Secret information processing	No pre-processing	Secret information segmentation with a square matrix
Synthetic texture size control	Synthetic texture capacity is equal or greater than the length of secret information	Synthetic texture capacity is equal to the length of secret information
Source texture embedding method	Embedding information in sparse and dense distribution	Embedding information with the seeded region growing method
Transmission of additional parameters	Cannot transmit together with synthetic texture	Transmit together with synthetic texture by PNG image features and LSB

5 Conclusions and future work

Our research makes improvement to steganography using reversible texture synthesis. This method is different from traditional steganography or texture synthesis. The texture is reorganized by using the seeded region growing algorithm, and the secret information is embedded in the process of synthesizing the texture. By using the LSB method, the important parameters are transferred together with the synthetic texture for the extraction of secret information. In further research, we will improve the selection strategy for the candidate blocks and increase the continuity of texture between the texture blocks.

Acknowledgments: This work was mainly supported by National Natural Science Foundation of China (No.61370218), Public Welfare Technology and Industry Project of Zhejiang Provincial Science Technology Department (No. 2016C31081, No. LGG18F020013)

References

Cox, I. J.; Kilian, J.; Leighton, F. T.; Shamoon, T. (1996): A secure, robust watermark for multimedia. *Information Hiding*, vol. 1174, pp. 185-206.

Gurusamy, R.; Subramaniam, V. (2017): A machine learning approach for MRI brain tumor classification. *Computers, Materials & Continua*, vol. 53, no. 2, pp. 91-109.

Huang, H.; Yang, S. (2011): Research on image information hiding technology based on LSB. *Office Automation*, no. 6, pp. 32-34 (In Chinese).

Niu, X.; Zhang, M. (2017): A study of invertible information hiding algorithm based on interpolation images. *Technology and Application in Web Security*, vol. 43, no. 12, pp. 120-129 (In Chinese).

Pan, L.; Qian, Z.; Zhang, X. (2016): Digital steganography based on constructive texture image. *Journal of applied science*, vol. 34, no. 5, pp. 625-632 (In Chinese).

Pradeep, A.; Mridula, S.; Mohanan, P. (2016): High security identity tags using spiral resonators. *Computers, Materials & Continua*, vol. 52, no. 3, pp. 185-195.

Qian, Z.; Zhou, H.; Zhang, W.; Zhang, X. (2017). Robust steganography using texture synthesis. *Advances in Intelligent Information Hiding and Multimedia Signal Processing*. Springer International Publishing, vol. 63, pp. 25-33.

Wu, K.; Wang, C. (2014): Steganography using reversible texture synthesis. *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, vol. 24, no. 1, pp. 130-139.

Zhang, W.; Liu, C. (2011). Public key and private key in electronic commerce. *Wealth of science and technology in China*, no. 12, pp. 234-236 (In Chinese).

Zheng, C.; Chen, J. (2010): A bit plane replacement information hiding algorithm based on RGB color space. *Information System Engineering*, no. 9, pp. 76-77 (In Chinese).

Zheng, S.; Wang, L.; Ling, B.; Hu, D. (2017): Coverless information hiding based on robust image hashing. *Intelligent Computing Methodologies*, vol. 10363, pp. 536-547.

Zhou, Z.; Cao, Y.; Sun, X. (2016): Coverless image information hiding based on image bag-of-words model. *Journal of applied science*, vol. 34, no. 5, pp. 527-536 (In Chinese).

Zhou, Z.; Sun, H.; Harit, R.; Chen, X.; Sun, X. (2015): Coverless image steganography without embedding. *International Conference on Cloud Computing and Security*, vol. 9483, pp. 123-132.