Reversible Texture Synthesis for Data Security

[1]Eshwari S. Mujgule, [2]N. G. Pardeshi
[1]PG Student, [2]Assistant Professor
[1]Computer Department,
[1]Sanjivani College of Engineering, Kopargaon, Kopargaon, India

***Abstract* - This dissertation work proposes Reversible Texture Synthesis an approach for data security. It uses the concept of patch which represents an image block of source texture where its size is user specified. A texture synthesis proce ss resamples a smaller texture image, which synthesizes a new texture image with a similar appearance and arbitrary size. The texture synthesis process is weaved into steganography to hide secret messages. In contrast to using an existing cover image to hide messages, the algorithm conceals the source texture image and embeds secret messages using the process of texture synthesis. This allows to extract the secret messages and source texture from a stego synthetic texture. The approach offers some advantages. First, the scheme offers the embedding capacity that is proportional to the size of the stego texture image. Second, the reversible capability inherited from this scheme provides functionality, which allows recovery of the source texture.**

***Index Terms* - Data embedding, patch, reversible, steganography, texture synthesis, stego synthetic texture.**

I. INTRODUCTION

Steganography [2] is practice of concealing a file, message, image or video within another file, message, image, or video. The steganographic application includes not openly acknowledged or displayed communication between two parties whose existence is unknown to possible attacker and whose success depends on detecting the existence of this communication [3]. Many of the image steganographic algorithms adopt an existing image as cover medium. The embedding of secret messages into the cover image can lead to image distortion in the stego image. This paper proposes Reversible Texture Synthesis an approach for data security. It uses the concept of pa tch which represents an image block of source texture where its size is user specified. A texture synthesis process resamples a smaller texture image, which synthesizes a new texture image with a similar appearance and arbitrary size. The texture synthesis process is weaved into steganography to hide secret messages. In contrast to using an existing cover image to hide messages, the algorithm conceals the source texture image and embeds secret messages using the process of texture synthesis. This allows to extract the secret messages and source texture from a stego synthetic texture. The approach offers some advantages. First, the scheme offers the embedding capacity that is proportional to the size of the stego texture image. Second, the reversible capability inherited from this scheme provides functionality, which allows recovery of the source texture.

Problem Statement

Reversible texture synthesis for data security uses the concept of steganography using reversible texture synthesis. The secret data is hidden into the texture image at sender side, it is done by generating patches from source texture and index table and composite image is generated, message is embedded and correct data can be recovered from the cover image with no change at receiver side. Major part of system will include Texture synthesis, message embedding and source texture recovery, message extraction and message authentication. The system is to be developed which will be easily embed into the different application where security is main concern.

II. LITERATURE SURVEY

Cohen *et al.* and Xu *et al.* [4], [5] uses the patch-based approach. Patch-based algorithms paste patches from a source texture instead of pixel to form the synthesized texture.

Advantage:

- This approach improves the image quality of pixel-based synthetic textures as texture structures inside the patches are maintained.

Disadvantage:

- Since patches are pasted with a small overlapped region during the synthetic process, one needs to make an effort to ensure that the patches agree with their neighbors.

Liang *et al.* [6] proposed the patch-based sampling and used the feathering approach for the overlapped areas of adjacent patches.

Advantage:

- The patch-based sampling algorithm is fast and it makes high-quality texture synthesis.

- The patch-based sampling algorithm works well for a wide variety of textures ranging from regular to stochastic.

Efros and Freeman [7] proposed a patch stitching approach called "image quilting for texture synthesis."
Advantage:
- Quilting is new, fast, yet very simple texture synthesis algorithm which produces good results for a wide range of textures.
- A dynamic programming technique is used to disclose the minimum error path through the overlapped region.

Ni *et al.* [8] introduces an image reversible data hiding algorithm which can recover the cover image without any distortion from the stego image after the hidden data have been extracted.
Advantages:
- Recover source texture without any distortion.

Ni et al. [9] proposed a general framework of current state of the art for reversible image data hiding .

III. SYSTEM OVERVIEW

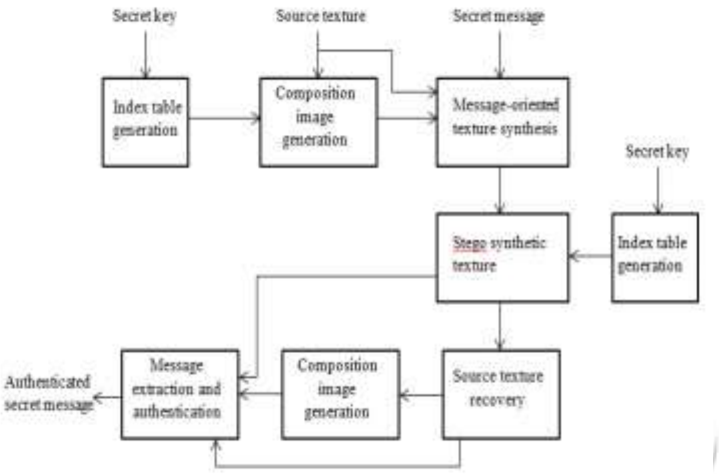


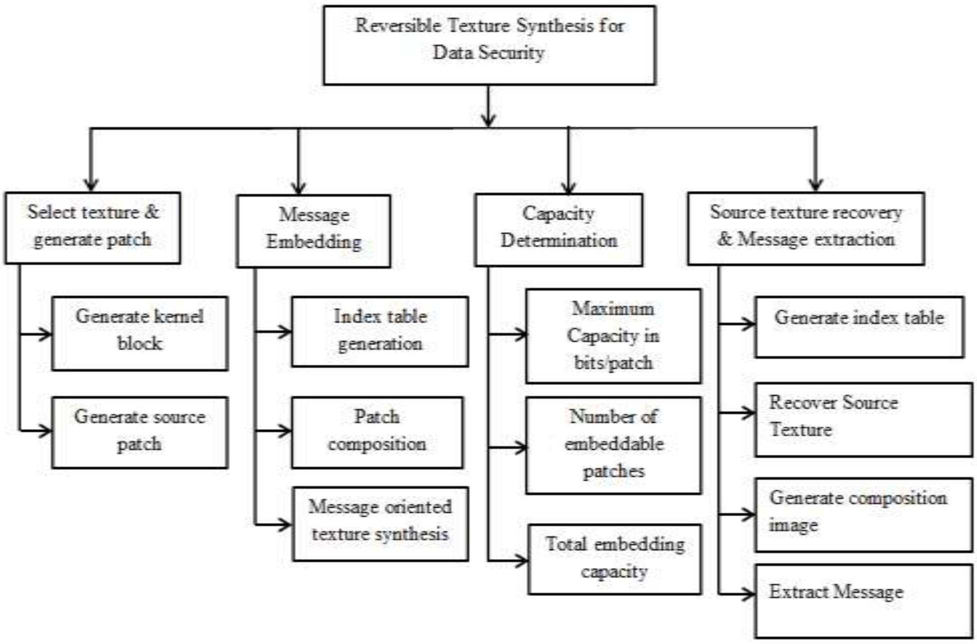
Fig .1: System Overview of Reversible Texture Synthesis for Data Security

Fig.1 Shows System Overview of Reversible Texture Synthesis for Data Security.
It consists of selecting texture and generating patches, message embedding, capacity determination, source texture recovery and message extraction.

Scope:
- Major parts of system will include Texture synthesis,Message embedding and Source texture recovery, Message extraction, Message authentication.
- To develop an system which can be easily embed in different applications where security is the main concern.
- To implement a system which will reduce overheads of text or image encryption algorithms.
- To develop a system which will retain the quality of service and the system performance.

Objective:
- To generate texture image patches.
- To generate Index table and Composite image.
- To embed message in the source texture without disturbing quality of the texture.
- To extract the data with no change

IV. BREAKDOWN STRUCTURE



Select Texture and Generate Patch Module

- Select the texture which is to be used as cover medium.
- Here the concept of patch is used. Patch is image block of source texture where its size is user specified. Patch size is denoted by its width and height.
- Then the concept of kernel block is used, which formed by subdividing the source texture into a number of non-overlapping kernel block each of which has size of kernel width and kernel height.
- Divide all the prediction errors into *L* clusters. The source patch is formed by expanding the kernel block with the depth at each side to produce a source patch.

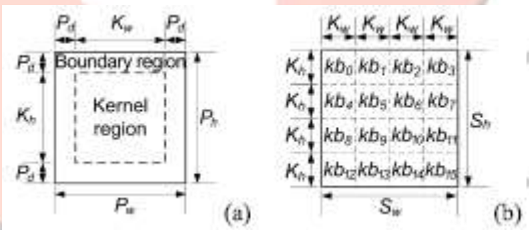


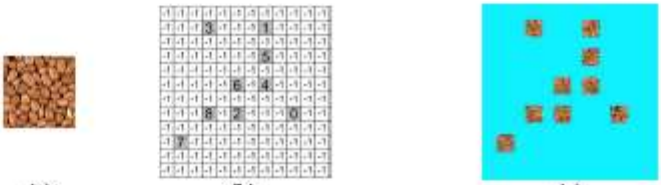
Fig .2 Patch, Kernel block and Source patch

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

Here, SPn is no of source patches, Sw and Sh is source texture width and height, Kw and Kh is kernel width and height.

Message Embedding Module

- The index table generation is carried out first where we produce an index table to record the location of source patch. The index table allows us to access the synthetic texture and retrieve the source texture completely. This reversible embedding style reveals one of the major benefits. The index table has the initial values of -1 for each entry, that shows the table is blank. We then assign values after distributing the source patch ID in the synthetic texture.
- In patch composition process we paste the source patches into the workbench to produce composite image.
- After generation of index table and composition image and after pasting source patches into workbench, we will embed the secret message via message oriented texture synthesis to produce stego synthetic texture.



(a) (b) (c)

Fig .3 Source texture, index table and composition image

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

TPn is number of patches in synthetic texture, Tw and Th is synthetic texture width and height, Pd is patch depth.

Capacity Determination Module

- Maximum capacity in bits/patch is calculated.
- Calculate number of embeddable patches which is difference between the number of patches in the synthetic texture and number of source patches subdivided from source texture.
- Calculate total embedding capacity which is product of maximum capacity in bits/patch and number of embeddable patches

Total Capacity= Bits per pixel that can be embedded in each patch X number of embeddable patches

Source Texture Recovery, Message extraction, and Message Authentication

- Generate the index table, given the secret key held in receiver side. The same index table as the embedding procedure can be generated.
- Recover the source texture. Source texture can be recovered or retrieved by referring the index table, then we arrange the blocks based on the order. Hence, the recovered texture will be same as the source texture.
- Generate the composite image, by pasting source patches into the workbench by referring the index table.
- Extract message by constructing candidate list and then perform match authentication step.

V. Result Analysis

Result of Embedding Capacity

Table I : Total Embedding capacity in bits

Pw * Ph= 58 * 58 Tw * Th= 500 * 500 Pd= 8					
Sh*Sw	**SPn**	**Epn**	**TC(5BPP)**	**TC(10BPP)**	**TC(BPPmax)**
128*128	9	87	437	875	1229
250*250	35	61	306	613	1518
300*300	51	45	229	458	1584

Table I shows total embedding capacity in bits that can be provided when different resolutions of the synthetic texture are produced by concealing various BPPs. It is important to point out that given a fixed number of BPP, the larger the resolutions of the source texture *Sw x Sh* (128 × 128 vs. 300 × 300), the smaller the total embedding capacity (TC). This is because the larger source texture will contain more source patches *SPn* (9 vs. 51) that we need to paste which cannot conceal any secret bits. This will reduce the number of embeddable patches (*EPn).*

Table II: Computing Time(Second)

Capacity	Pure	5BPP	10BPP	BPPmax
128*128	780	1200	1325	1262
250*250	931	1580	1622	1600
300*300	1100	1820	2050	2112

Image Quality Comparison

1. **Mean Squared Error for Overlapped area (MSEO)**

Table III Comparison of MSEO with respect to Embedding Capacity

	Pure	5BPP	10BPP
Rope net	6842	6846	6790
Metal	8763	8768	8919
Peanuts	2832	2837	2913
Ganache	1116	1125	1204

Mechanism to determine image quality is MSEO. The MSEO has a non-zero value even in the case of the pure patch based texture synthesis. If the MSEO produces a small value, it implies that the synthetic texture shows a high image quality of the overlapped areas. Obviously, the lower the MSEO value, the higher quality of the synthetic texture image.

2. **Structural SIMilarity index (SSIM)**

Table IV Comparison of SSIM Index

Pure vs.	5BPP	10BPP
Rope net	0.0191	0.0305
Metal	0.0161	0.0012
Peanuts	0.032	0.052
Ganache	0.0201	0.0673

SSIM (Structural SIMilarity) index used to quantify the similarity between the pure and stego synthetic textures. The SSIM is an image quality assessment method for measuring the change in luminance, contrast, and structure in an image. The SSIM index is in the range of [−1, 1] and when it equals to 1, the two images are identical.

VI. Conclusion

This paper proposes the Reversible texture synthesis for data security. The source texture is given and the scheme produces the stego synthetic texture can hiding the secret message. The patch based concept is used instead of pixel based approach. Provides the reversible approach to recover source texture and secret message from stego synthetic texture.

VII. Acknowledgment

"Reversible Texture Synthesis for data Security" has been a wonderful subject to research upon, which leads ones mind to explore new heights in the field of Computer Engineering. I dedicate all my works to my esteemed guide, Prof. N. G. Pardeshi, whose interest and guidance helped me to complete the work successfully. This experience will always ssteer me to do my work perfectly and professionally. I also extend my gratitude to Prof. D. B. Kshirsagar (H.O.D. Computer Engineering Department) and Prof. P. N. Kalvadekar (P. G. Cordinator) who has provided facilities to explore the subject with more enthusiasm. I express my immense pleasure and thankfulness to all the teachers and staff of the Department of Comp. Engg., S.R.E.S COE, Kopargaon for their co-operation and support. Last but not the least, I thank all others, and especially my friends who in one way or another helped me in the successful completion of this dissertation.

REFERENCES

[1]. Kuo-Chen Wu and Chung-Ming Wang, ”Steganography Using Reversible Texture Synthesis”, IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 24, NO. 1,(2015).
[2]. N. F. Johnson and S. Jajodia, “Exploring steganography: Seeing the unseen,” *Computer*, vol. 31, no. 2, pp. 26–34, 1998.
[3]. F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, “Information hiding survey,” Proc. IEEE, vol. 87, no. 7, pp. 1062–1078, Jul. 1999.
[4]. 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.
[5]. K. Xu *et al.*, “Feature-aligned shape texturing,” *ACM Trans. Graph.*, vol. 28, no. 5, 2009, Art. ID 108.
[6]. 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.
[7]. 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.
[8]. 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.
[9]. X. Li, B. Li, B. Yang, and T. Zeng, “General framework to histogram-shifting-based reversible data hiding,” *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2181–2191, Jun. 2013.