Comments and Corrections

Comments on "Steganography Using Reversible Texture Synthesis"

Hang Zhou, Kejiang Chen, Weiming Zhang, and Nenghai Yu

***Abstract*— Message hiding in texture image synthesis is a novel steganography approach by which we resample a smaller texture image and synthesize a new texture image with a similar local appearance and an arbitrary size. However, the mirror operation over the image boundary is flawed and is easy to attack. We propose an attacking method on this steganography, which can not only detect the stego-images but can also extract the hidden messages.**

***Index Terms*—Steganography, steganalysis, data hiding, texture image, mirror operation.**

I. Introduction

THE above paper [1] proposes a method of steganography using reversible texture synthesis. In this section, we first give some notation and brief review of the paper, and then we point out the vulnerability of steganography in images that are dealt with by a mirroring operation over their boundaries. Everywhere in this paper, vectors will be always typeset in boldface lower case, while we reserve the blackboard style for matrices.

We denote the source image by **A**, the synthetic image by **S**, and the embedded message by **m**. A patch represents a user-specified block of the source image, the size of which is denoted by $P_w \times P_h$. As shown in Fig. 1(a), a patch contains the central kernel region with a size of $K_w \times K_h$ and the boundary region with a depth of P_d. We denote the size of **A** by $S_w \times S_h$ and the size of **S** by $T_w \times T_h$.

The course of reversible image synthesis proceeds as follows. First, divide a source image **A** into same-sized non-overlapped kernel blocks of the same size, which sum up to SP, where

$$SP = n_w \times n_h = \frac{S_w}{K_w} \times \frac{S_h}{K_h}. \tag{1}$$

A kernel-centered expansion with a depth of P_d operates as illustrated in Fig. 1(b). The expansion on the boundary of **A** is implemented with a mirroring operation. To synthesize an image with a given size, a random padding step is first carried out by employing the total source patches with a user-specific secret key. A sliding window is then employed with a stride size of one pixel, following the scan-line order to create candidate patches to pad into synthetic image **S**. The number of candidate patches CP shall be acquired via

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

and the number of patches in **S** is acquired by

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

The process of padding is a zigzag pattern for message embedment. Patches generated by expansion or mirroring are padded randomly in **S**, which is shown in Fig. 1(c). Since overlapped regions exist when iteratively padding a candidate patch to one blank space in **S**, a descending sort of the mean square error (MSE) of the overlapped region between the candidate patch and synthesized area is obtained. The smaller the MSE, the more similar the candidate is to the synthesized area. After producing the rank table above, the decimal

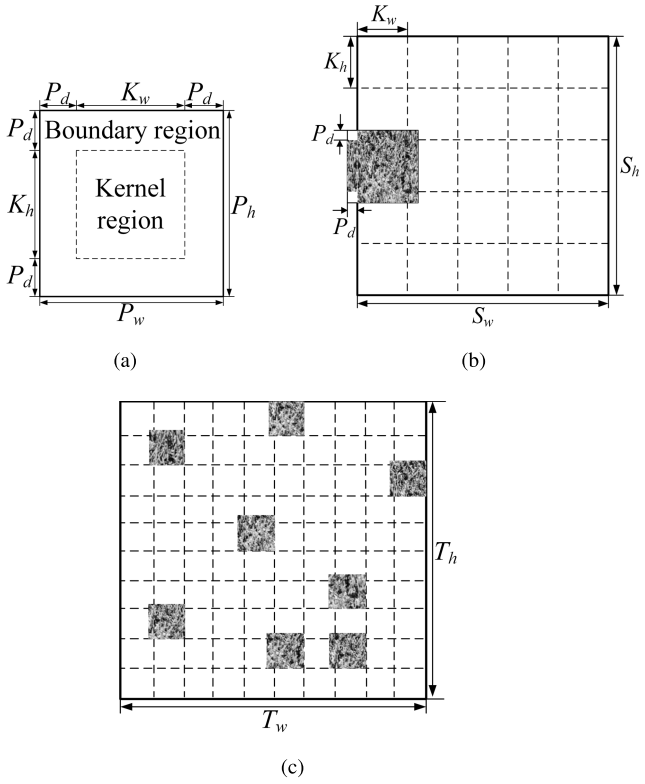

Fig. 1. The structure of patches and kernels in a source image **A** and a synthetic image **S**. (a) A patch consists of kernel and boundary regions. (b) The diagram of source patches generated by expanding or mirroring boundary using kernel blocks in a source image. (c) The synthetic image after a padding step of source patches.

Manuscript received February 5, 2016; revised July 20, 2016; accepted January 19, 2017. Date of current version February 17, 2017. This work was supported in part by the Natural Science Foundation of China under Grant U1636201, Grant 61572452, and Grant U1536108. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Charles Boncelet. *(Corresponding author: Weiming Zhang.)*

The authors are with the CAS Key Laboratory of Electromagnetic Space Information, University of Science and Technology of China, Hefei 230026, China (e-mail: zh2991@mail.ustc.edu.cn; chenkj@mail.ustc.edu.cn; zhangwm@ustc.edu.cn; ynh@ustc.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2657886

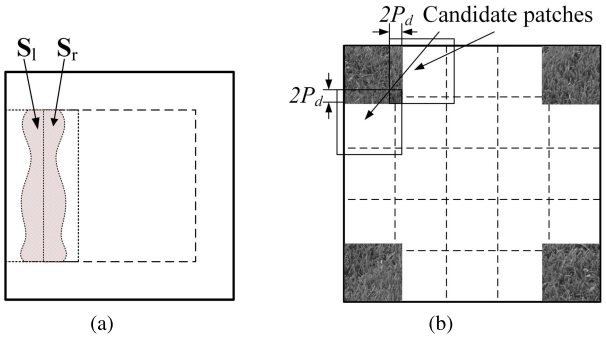

Fig. 2. (a) Seam line is depicted. $\mathbf{S}_\mathrm{r}$ is the right area of the kernel periphery, and $\mathbf{S}_\mathrm{l}$ originates from $\mathbf{S}_\mathrm{r}$. (b) The process of recovering source images. Since corner patches are easy to get, we pad them first. In accordance with the MSE of the overlapped region, patches below or on the right side of the last patch are then padded. The width of the overlapped region is $2P_d$.

number of a n-bit embedding message decides the selection of candidate patch whose rank equals the value of message. An image quilting technique [2] is adopted to reduce the visual artifact during the synthesis period.

The method proposed by Wu and Wang [1] has one severe problem due to the mirroring operation on the boundary of the source images, which may divulge the information of source images and embedded messages.

II. Proposed Method

In this section we propose a method to recover the source image $\mathbf{A}$ by breaking down the synthetic texture image $\mathbf{S}$. We denote the recovered source image by $\mathbf{A}'$, by which we can simulate the process of synthesizing image $\mathbf{S}$. In this synthetic process, we can analyze whether $\mathbf{S}$ is a stego image and extract the hidden messages from the stego image.

A. Source Image Recovery

Due to the mirroring operation, synthetic images can be broken down. By comparing the similarity of an expanded boundary area with an inside boundary area, we can search patches in the synthetic image $\mathbf{S}$ and identify the patches coming from the four corners and boundaries of source image $\mathbf{A}$. To do that, we define a metric on similarity below.

Since the overlapped region contains two parts of different patches, the selection of pixels depends on the image quilting technique. According to the seaming method [2], we assume that we are sewing an overlapped area between the left patch and right patch, as seen in Fig. 2(a). Suppose we are treating a patch from the left boundary of a source texture image $\mathbf{A}$. Denote the left area of the kernel periphery by $\mathbf{S}_\mathrm{l}$, the right area by $\mathbf{S}_\mathrm{r}$, and the mirror of $\mathbf{S}_\mathrm{l}$ by $\mathbf{S}'_\mathrm{l}$. Owing to the mirroring operation over the boundary of $\mathbf{A}$, the shape of the texture in $\mathbf{S}'_\mathrm{l}$ is similar to that of $\mathbf{S}_\mathrm{r}$. As for other patches that come from the inner part, upper boundary, bottom boundary, or right boundary, the similarity is comparatively low.

Denote the similarity of the overlapped region by w_p. Since it is difficult to find seam lines, a straightforward counting of identical pixel values between $\mathbf{S}'_\mathrm{l}$ and $\mathbf{S}_\mathrm{r}$ can represent w_p. It is more effective if we assign a weight factor to each pixel. Since pixels closer to the kernel periphery are more likely to be derived from the original patch, an empirical factor setting is defined as $h(i)$, where i is the position of a pixel in the overlapped region and $1 \leq i \leq P_d$. For each overlapped region, we label pixels on the left side of the seam line as 0 and pixels on the right side of the seam line as 1,

TABLE I
Weight h for Different i

i Subscript	1	2	3	4	5	6	7	8
Optimal h	1	1.66	2.37	3.07	3.77	4.43	4.96	5.34

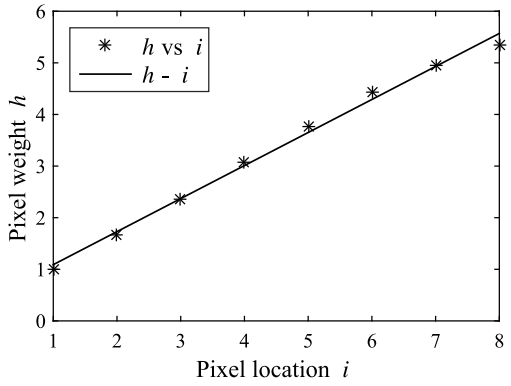

Fig. 3. The scatter plot of (i, h) for improving weight selection of pixels.

and then sum up the labels of overlapped regions at each i from randomly selected 100 synthetic images. We perform normalization of each $h(i)$ by dividing the sum of label at $i = 1$, which is taken as weight.

We list the weight $h(i)$ for different i in Table I, and make a scatter plot in Fig. 3. We discover that the optimal $h(i)$ increases with i, and establish a linear regression model between $h(i)$ and i,

$$h(i) = 0.64i + 0.45, \tag{4}$$

by which we can estimate the optimal $h(i)$ for a given value of i.

Thus, a refined construction of w_p is defined as follows:

$$w_p = \frac{1}{LP_d} \sum_{i=1}^{L} \sum_{j=1}^{P_d} h(j) I(\mathrm{S}'_\mathrm{l}(i, j) = \mathrm{S}_\mathrm{r}(i, j)), \tag{5}$$

where L is the height of the overlapped region, and I is an indicator function.

Therefore, every patch from the synthetic image $\mathbf{S}$ gets a corresponding value w_p. The value w_p of patches at left boundary of the source texture image $\mathbf{A}$ should appear to be quite larger than other values. We sort w_p over patches in descending order. The patch of top order is selected to recover some portion of source image. To recover the source image $\mathbf{A}'$, we pad selected patches onto a blank $\mathbf{A}'$ which is much like solving jigsaw puzzles. Summing up two mirroring areas rather than one makes it more distinct to find out four corner patches, thus firstly we pad the corners, as seen in Fig. 2(b).

After that, we pad image boundaries. We go about the padding procedure in a left-to-right and top-to-bottom design. Given all of the candidate patches, we ought to find the most suitable one to pad the place below the patch from the top left corner. An overlapped region with a width of $2P_d$ occurs when seeking the optimal solution, as described in Fig. 2(b). Therefore, we replace P_d with $2P_d$ in Eq. (5). The optimal patch corresponding to the highest rank of similarity is padded afterwards.

Once we have padded a ring-shaped boundary of the source image $\mathbf{A}'$, we perform a similar iterative operation, dealing with corner patches and boundary patches until all padding places are filled.

As for attackers, it is necessary to obtain the size of kernels and P_d. Given T_h and T_w of a synthetic image $\mathbf{S}$, we are able to acquire several possible candidates of K_h, K_w and P_d. With a specific case,

we compute w_p of each patch, and find the maximum. We then traverse all the cases and find the maximal w_p, which corresponds to the matched size.

In the process of recovering the source image, we record the position of each source patch at the the synthetic image $\mathbf{S}$, which will be used to extract the message.

B. Message Extraction

As we have obtained a recovered source image $\mathbf{A}'$, we then generate candidate patches from $\mathbf{A}'$ through a sliding window the same as in Wu's method [1]. With these candidate patches, we can simulate the process of synthesizing image $\mathbf{S}$. First and foremost, pad each source patch generated by expansion or mirroring operation from $\mathbf{A}'$ onto a new synthetic image $\mathbf{S}'$ according to its position on the synthetic image $\mathbf{S}$. And then, padding candidate patches in a zigzag pattern, each time we have to compute the MSEs of an overlapped region between the current patch $\mathbf{B}_\mathrm{C}$ and candidate patches, we generate a sequence of MSE values $\mathbf{v}$ in descending order. On the other hand, we calculate the MSE v of overlapped region between the electee in $\mathbf{S}$ with the patch $\mathbf{B}_\mathrm{C}$. By observing the position of v in the sequence $\mathbf{v}$, we extract the message carried on this electee. Cascading together the messages extracted from each patch, we get an estimated version $\mathbf{m}$, denoted by $\mathbf{m}'$. If $\mathbf{m}'$ is an all-zero sequence, the image $\mathbf{S}$ was't embedded with messages; otherwise $\mathbf{S}$ is a stego image and we take $\mathbf{m}'$ as the extracted message.

Yet there is a small probability of collapse when extracting messages. Due to the fact that some candidate patches may be identical to patches generated by expansion that carry no message, during message extraction phase, it may be difficult to tell the two patches apart, which causes false message extraction. One of the improvement on the security of Wu's method that we put forward is that the sender generates repetitive patches by expansion without messages, which are padded onto synthetic images. As for attackers, despite recovering source image correctly, redundant repetitive patches that carry no message will be considered candidate patches that carry message, thus affect the extraction of message.

III. Experiment Results

Experiments were carried out with Brodatz Textures [3]. Since texture images are insufficient, we resized and clipped these images to constitute 174 source images with size of 128×128. We generated 2,088 cover synthetic images and an identical number of stego synthetic images by embedding messages with relative payloads varying from 1 bit per patch (bpp) to 12 bpp. We set $P_d = 8$, $P_w = P_h = 32$ and $T_w = T_h = 488$, which are same to Wu's configuration. From the total 4,176 synthetic images, we detected the stego images and extract messages.

TABLE II
Performance of Proposed Steganalysis Method

Recovery Rate	Missing Alarm (P_{MA})	False Alarm (P_{FA})	Correct Message Extraction Rate
96.77%	0	5.71%	94.66%

We first estimate the size of K_w, K_h and P_d. Possible cases are stated: $\{(K_w, K_h, P_d) \mid (32, 32, 8), (18, 18, 4), (16, 16, 12), (43, 43, 16)\}$. From the experiment, a high detection rate with 98.12% of synthetic images can be determined with $K_w = K_h = 32$ and $P_d = 8$.

In Table II, the recovery rate denotes the probability of the source image being correctly recovered. A missing alarm (P_{MA}) denotes the ratio of stego images being identified as cover images (i.e. synthetic images without embedment of messages), while a false alarm (P_{FA}) is the error ratio of wrongly assigning a cover image to the set of stego images. It is shown that 96.77% of source images can be losslessly recovered, and we did not miss any stego images, with P_{FA} being as low as 5.71%. In addition, the extraction rate, defined as the ratio of correctly extracted messages to total messages, is higher than 94%.

IV. Conclusion

We proposed a method to successfully attack the synthetic image based steganography [1]. The security of Wu's method [1] can be improved. Since the mirroring operation will leak information of patch scrambling, it is crucial to find a substitute for mirroring while ensuring the quality of texture images. One solution is to find a proper texture area in the assemblage of candidate patches and replace the mirroring area.

References

[1] K. C. Wu and C. M. Wang, "Steganography using reversible texture synthesis," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 130–139, Jan. 2015.

[2] 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.

[3] (1997). *Brodatz Texture Database*. [Online]. Available: http://multibandtexture.recherche.usherbrooke.ca/original_brodatz.html.