
# Targeted Attack and Security Enhancement on Texture Synthesis Based Steganography

Hang Zhou, Kejiang Chen, Weiming Zhang, Zhenxing Qian, Nenghai Yu

PII: S1047-3203(18)30087-7
DOI: https://doi.org/10.1016/j.jvcir.2018.04.011
Reference: YJVCI 2178

To appear in: *J. Vis. Commun. Image R.*

Please cite this article as: H. Zhou, K. Chen, W. Zhang, Z. Qian, N. Yu, Targeted Attack and Security Enhancement on Texture Synthesis Based Steganography, *J. Vis. Commun. Image R.* (2018), doi: https://doi.org/10.1016/j.jvcir.2018.04.011

# Targeted Attack and Security Enhancement on Texture Synthesis Based Steganography

Hang Zhou[a], Kejiang Chen[a], Weiming Zhang[a,*], Zhenxing Qian[b], Nenghai Yu[a]

[a]*CAS Key Laboratory of Electromagnetic Space Information, University of Science and Technology of China, Hefei 230027, China.*
[b]*School of Computer Science, Fudan University, 201203, China.*

**Abstract**

We describe an effective and efficient strategy building steganography detector for patch synthesis based steganography, one case of which is reversible texture synthesis based steganography method proposed by Wu *et al.* [12]. By exploiting the observation that steganography destroys optimization of matching extent between the synthetic patch and optimal candidate patch, we reconstruct the two patches from an overlapped region to extract the existence of optimality, which are distinct between cover and stego images, to form features. Support vector machine (SVM) is implemented for classification. Meanwhile, a variant of Wu *et al.*'s steganographic method is proposed with reinforced security, by padding redundant regions carrying no message around the periphery of the synthesized image and generating additional candidate patches to increase capacity. Experiments demonstrate that the modified algorithm offers not only better resistance against the state-of-the-art steganalysis methods and steganalytic attack we developed, but also a larger embedding capacity.

*Keywords:* Texture image, steganalysis, texture synthesis, steganography.

## 1. Introduction

Steganography is a technique for covert communication and privacy protection, which is now a fairly standard concept in computer science. The process of modern steganography is that a steganographic system embeds hidden content in unremarkable cover media so as not to arouse the suspicion of an eavesdripper [1].

Currently, the majority of image steganographic methods adopt natural images as cover images to embed data, where the most successful approach to design content adaptive steganography is based on minimizing the distortion between the cover and the corresponding stego object, which is acquired by assigning a cost

*Corresponding author. Tel.: +86 0551 3600683
*Email addresses:* zh2991@mail.ustc.edu.cn (Hang Zhou), chenkj@mail.ustc.edu.cn (Kejiang Chen), zhangwm@ustc.edu.cn (Weiming Zhang), zxqian@fudan.edu.cn (Zhenxing Qian), ynh@ustc.edu.cn (Nenghai Yu)
[1]This work was supported in part by the Natural Science Foundation of China under Grant U1636201, Grant 61572452, Grant U1536108, Grant 61572308 and Grant U1736213.

of changing each cover element. Syndrome-Trellis Codes (STCs) [2] are used to embed messages after minimizing the total distortion as a sum of costs of all modified elements. The principle of distortion's definition is that pixels that are easily modelable in regions should be assigned high costs. Methods such as HUGO [3], WOW [5], UNIWARD [7], HILL [8], MiPOD [9] and CPP [23] are brought up successively based on the principle.

However, steganography may be attacked by steganalysis which aims to expose the presence of hidden data. In general, steganalytic approaches are classified into two categories: specific and universal. The former detects the presence of a message embedded by a particular steganographic algorithm, while the latter targets at message detection on comprehensive steganographic algorithms with varying embedding strategies. As for universal image steganalyzers, much have been well-studied in the literature. It is noteworthy that since plenty of practical steganographic algorithms perform embedding by applying a mutually independent embedding operation to all or selected elements of the cover, the effect of embedding is equivalent to adding to the cover an independent noise-like signal called the stego noise [4]. Steganalyzer's features are usually generated by exploiting correlations between the predicted residuals of neighboring pixels[10]. Fridrich *et al.* [17] and Ker [18] propose methods specifically for the detection of LSB replacement. Early feature-based steganalysis algorithms used only a few dozen features, e.g., 72 higher order moments of coefficients obtained by transforming an image using quadratic mirror filters [16]. The SPAM [4] set for the second-order Markov model of pixel differences has a dimensionality of 686. Whereafter, SRM [6] is proposed with 34,671 dimensions to have a better performance in steganalysis, and maxSRM [25] forms the co-occurrence matrices considering the maximum estimated modification probability of a group of pixels as a weight coefficient, for which the steganalytic feature is inclined to extract features from targeted region.

Since the demand for synthetic texture images boosts greatly with the development of computer graphic, applications of which include online games, cinefex, 3D roads, virtual reality, etc., texture images can serve as favorable carriers for secret message. The first attempt to design texture synthesis based steganography appeared in [19, 20] by Otori and Kuriyama with pixel-based texture synthesis combining data coding. Secret messages are encoded into colored dotted patterns picked from textures and they are directly painted on a blank image. The rest of the pixels are filled using pixel-based texture synthesis method, where the capacity is determined by the dotted patterns. Wu *et al.* [12] proposed a reversible texture synthesis based steganography method, which resamples a smaller texture image and synthesizes a new texture image with a similar local appearance and an arbitrary size. Message is embedded by the selection of candidate regions generated from the source image. Qian *et al.* [21] proposed a robust steganography that can counter JPEG compression at the cost of low capacity.

The design of texture image features is more challenging since it has similar complexity between cover and stego texture images. As far as we know, steganography will break down the correlation among adjacent pixels, and it is more noticeable to find the modifications in smooth areas than in textural areas after

steganography. Therefore, it will be less effective for steganalysts to extract prominent features of pixels in texture areas. Zhou *et al.* [22] proposed a specific steganalytic algorithm on Wu's method by inspecting mirroring region and reconstructing the original texture image. However, steganographers can fix the flaw by finding a substitute for mirroring region to avoid information leakage during steganography. To the best of our knowledge, there is no literature that related texture image with steganalysis.

The revised method of Wu [12] represents state-of-the-art texture steganography. We further analyze the approach [12] through specific steganalysis to evaluate its security, explore possible security holes and put forward security-enhanced steganography on texture images.

In this paper, we propose a specific steganalytic algorithm for determining whether the synthesized image generated by the method proposed by Wu *et al.* contains message, which is now a superior steganographic algorithm on texture images. Such steganography makes the optimality between two adjacent patches to be synthesized drop to a certain extent, while it is expected to be optimal without steganography. We have improved the expression by the following modifications: Specifically, we reconstruct the original adjacent two patches from synthesized regions, and exploits the suitability degree of matching between them to conduct steganalysis. Thus we coin a new acronym ReSid standing for **Re**constructed **Si**milarity **D**egree detector.

We also propose a security-enhanced texture steganographic algorithm with improved undetectability and larger capacity over Wu *et al.*'s. The new algorithm pads redundant regions carrying no message around the periphery of the synthesized image by identical image quilting technique [13]. It is nearly impossible for attackers to estimate the sizes of redundant regions and determine the actual size of synthesized image and then implement steganalysis algorithm that we put forward. We even improve the maximum capacity of a single patch by generating more candidate patches to form a larger candidate set. Experimental results show that the proposed addition of redundant regions offers improved performance against the proposed steganalytic attack and traditional state-of-the-art steganalytic methods.

The rest of the paper is organized as follows: Section II starts with some notations, a brief review of Wu *et al.*'s texture synthesis based steganography and proposed steganalytic method. Section III illustrates the security-enhanced steganographic algorithm. Some implementation issues and performance comparison are discussed in Section IV. The conclusions and further directions are drawn in Section V.

## 2. Steganalytic Algorithm on Wu *et al.*'s Method

In this section, we first briefly describe the texture steganographic algorithms by Wu *et al.* [12] and then present in detail the proposed steganalytic method against the algorithm. Throughout this paper, the calligraphic font will be used solely for sets. Vectors will be always typeset in boldface lower case, while we reserve the blackboard style for matrices (e.g., $A_{ij}$ is the $ij$th element of matrix $\mathbf{A}$).

### 2.1. Wu et al.'s Texture Synthesis Steganography

This section contains an overview of candidate sorting based steganographic algorithm using texture synthesis [12]. We denote the source image by $\mathbf{A}$, the synthetic image by $\mathbf{S}$ and the embedded message by $\mathbf{m}$. 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 Figure 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 $\mathbf{A}$ by $S_w \times S_h$ and the size of $\mathbf{S}$ by $T_w \times T_h$.

The course of steganography is elaborated as follows. First, divide $\mathbf{A}$ into same-sized non-overlapped kernel blocks. A kernel-centered expansion with a depth of $P_d$ is operated, as illustrated in Figure 1(b). The four boundaries of a patch are replicated from the nearby kernels. The expansion on the boundary of $\mathbf{A}$ 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, as shown in Figure 1(c). The number of patches $n_T$ in $\mathbf{S}$ is acquired by

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

And then, in $\mathbf{A}$, a sliding window is employed with stride size of one pixel following the scan-line order to create candidate patches to pad into $\mathbf{S}$. The number of candidate patches $n_C$ are derived by

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

where each candidate patch is marked with a sequence number.

Image quilting technique [13] is adopted to reduce the visual artifact during the synthesis period, which targets to find a seam line between two blocks on the pixels where the two textures match best. We denote two regions by $\mathbf{B}_l$ and $\mathbf{B}_r$ that overlap along their vertical direction respectively, as shown in Figure 1(e). The synthesized region is called an **O**ver**L**apped **R**egion (abbreviated to OLR). Let $\mathbf{D}$ denote some perceptual distance between two patches, which is a normalized sum of squared differences metric. Denote vertical seam line by $\mathbf{q} = \{q_j | j = 1, 2, ..., K_h\}$, and the minimal one $\hat{\mathbf{q}}$ is acquired by

$$\begin{aligned} \hat{\mathbf{q}} &= \arg\min_{\mathbf{q}} \sum_{j=1}^{K_w} D(q_j, j)^2 \\ s.t. &|q_j - q_{j-1}| \leq 1. \end{aligned} \tag{3}$$

The shortest path problem can be solved by dynamic programming algorithms. Similar procedure can be applied to horizontal overlaps.

Virtually the process of padding is a zigzag pattern for message embedment, as is shown in Figure 1(d). Since there exist OLRs when padding a candidate patch to one blank space in $\mathbf{S}$ in an iterative way, descending sort of the mean square error (MSE) of the OLRs between the candidate patch and synthesized area is obtained to form a rank table. The smaller the MSE, the more similar the candidate is to the

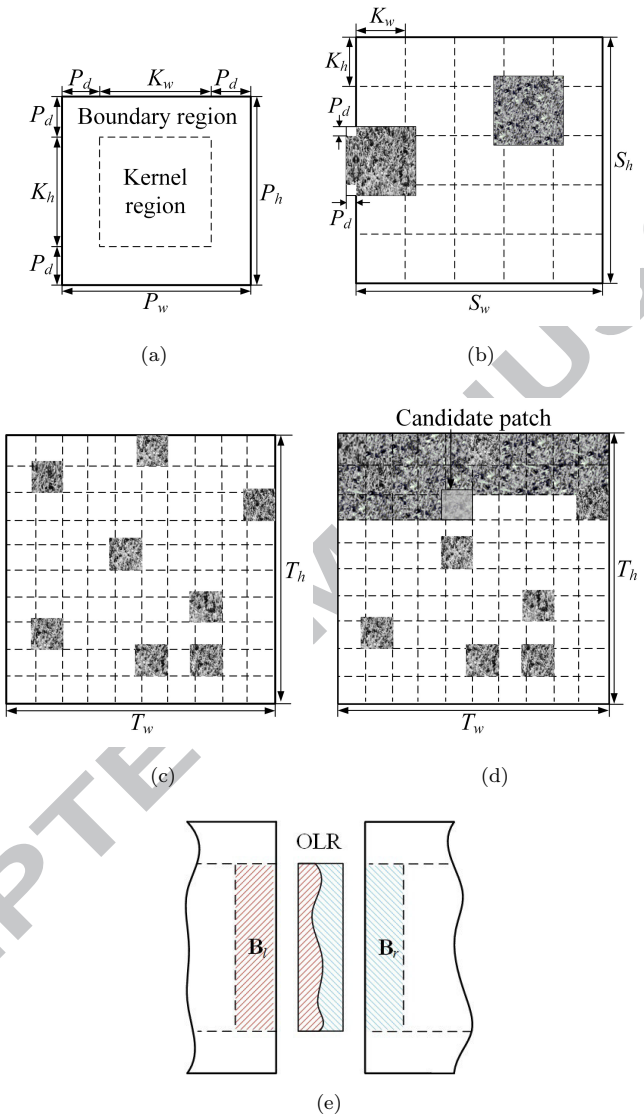



Figure 1: The structure of patches and kernels in a source image **A** and a synthetic image **S**. (a) A patch consists of a kernel and boundary regions. (b) Source patches generated by expanding or mirroring the boundary regions of kernel blocks in **A**. (c) **S** after a random padding step. (d) Zigzag padding pattern of synthesizing **S**. (e) Two regions $\mathbf{B}_l$ and $\mathbf{B}_r$ from two patches to be synthesized together using image quilting technique. The OLR represents the spliced region.

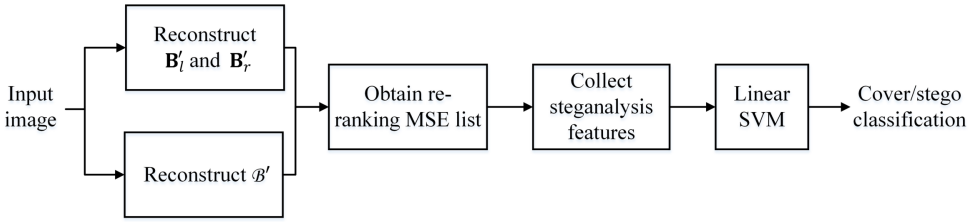



Figure 2: An overview of feature extraction and steganographic texture image detection flowchart. $\mathbf{B}'_l$, $\mathbf{B}'_r$ and $\mathcal{B}'$ are reconstructed from OLRs to simulate the process of image synthesis. We then obtain a re-ranking MSE list to get a corresponding rank. Traversing all the OLRs, we aggregate all the ranks to form a rank set. Features of the proposed steganalyzer are extracted from the rank set, followed by a linear SVM for cover/stego image classification.

synthesized area. After we produce the rank table above, the decimal number of the embedding message decides the selection of the candidate patch whose rank equals the value of message.

As for the receiver side, a legal recipient can recover $\mathbf{A}$ with the secret key. By simulating the process of synthesizing $\mathbf{S}$, candidate patches and a new synthetic image $\mathbf{S}'$ are generated. In a zigzag way of padding candidate patches onto $\mathbf{S}'$, each time we compute the MSEs of OLR between the current patch and candidate patches, and generate a sequence of MSE values in descending order. On the other hand, we calculate the MSE of the OLR between the electee in $\mathbf{S}$ with the current patch. By observing the position of the patch in the sequence, we extract the message carried on this electee. Thus we retrieve message $\mathbf{m}$. As for attackers without key, they cannot recover $\mathbf{A}$, hence can hardly get any message directly.

For brevity, Wu *et al.*'s method, **Ca**ndidate **So**rting based texture synthesis steganography is abbreviated to CASO.

### *2.2. Proposed Steganalytic Algorithm*

CASO is insecure for its construction, which may be attacked by eavesdroppers. Since steganography destroys the optimization of matching degree between the synthetic patch and optimal candidate patch, by reconstructing the two patches from synthesized images and extracting the existence of optimality, we can conduct efficient steganalysis. It is noteworthy that five shapes[1] of OLR occur in $\mathbf{S}$, where each shape can be decomposed to several rectangle regions. If every rectangle region is the optimized matched patch, then the shape possesses optimality. What it comes down to is to capture the existence of optimality between two rectangle patches.

CASO employs $\mathbf{B}_l$ as the fixed region to be synthesized and $\mathbf{B}_r$ selected from sorted candidate patch list $\mathcal{B} = \{\mathbf{B}_1, \mathbf{B}_2, ..., \mathbf{B}_{n_C}\}$ with MSE metric. The chosen $\mathbf{B}^*$ is decided by the rank of sorted MSEs equal the value of the message. Suppose that attackers are able to extract the hidden message that equals the

[1]The five shapes of OLR are not shown for lack of space (see [12]).

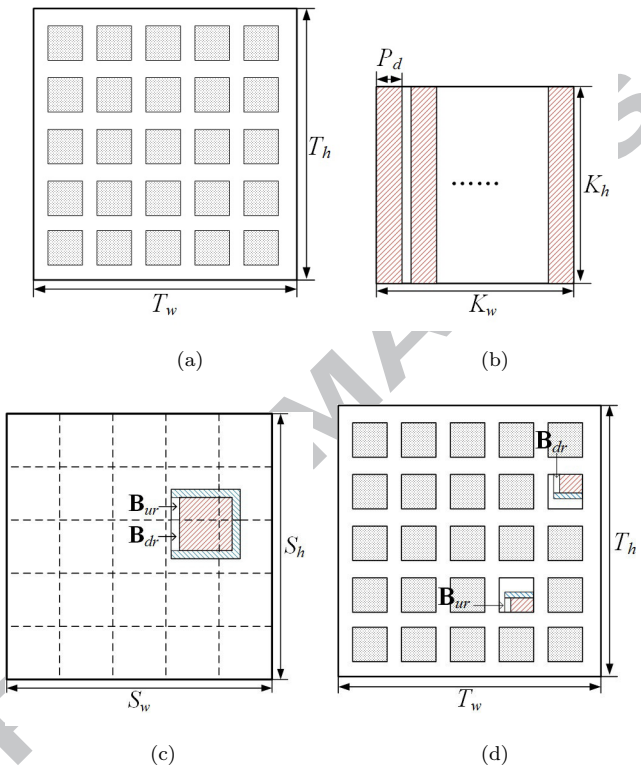



Figure 3: (a) Structure of synthetic images. Squares represent kernels that are not modified. (b) A sliding window with a size of $K_h \times P_d$ and a stride size of one pixel in a kernel. (c) The chosen candidate patch $\mathbf{B}^*$ is located over 4 kernel regions in a source image. (d) $\mathbf{B}_r$ is separated into two parts ($\mathbf{B}_{dr}$ and $\mathbf{B}_{ur}$) in a synthetic image.

rank given the same $\mathbf{B}_l$ and $\mathcal{B}$, we aim at recovering $\mathbf{B}_l$ and $\mathcal{B}$. However, it is impossible to reconstruct the process of synthesization losslessly. For attackers, since they do not own the key to find the source patches generated by expansion or mirroring, potential $\mathcal{B}$ cannot be recovered to the original version; since $\mathbf{B}_l$ and $\mathbf{B}_r$ have been synthesized together, they cannot recover $\mathbf{B}_l$ and $\mathbf{B}_r$ directly. Next we describe the recovery of $\mathcal{B}$ and $\mathbf{B}_l$ individually.

It is clear that $\mathcal{B}$ can only be generated from $\mathbf{A}$; without key, $\mathcal{B}$ can no longer be generated. Thus we use a substitute to approximate $\mathcal{B}$ to reproduce the process of MSEs' numeration, denoted by $\mathcal{B}'$. A sliding window with a size of $K_h \times P_d$ (assume that $K_h$ and $P_d$ are given) and a stride size of one pixel is employed in all the kernels collected from $\mathbf{S}$, producing large amount of candidate regions, which represent the approximation of candidate patches, as shown in Figure 3(a, b). We denote the number of candidate $\mathcal{B}'$ patches by $n'_C$. It is evident that a larger relative payload will cause a larger rank in the statistic point of view, making the substitution feasible.

Most of $\mathbf{B}_l$ and $\mathbf{B}_r$ can be recovered, yet not directly. The keystone and difficulty of this paper are the reconstruction of $\mathbf{B}_l$ and $\mathbf{B}_r$. Owing to the fact that a sliding window moves over $\mathbf{A}$ with a stride of one pixel, the chosen candidate patch $\mathbf{B}^*$ may be located over 4 kernel regions in most instances, which is shown in Figure 3(c). The left boundary $\mathbf{B}_r$ of the patch is made up of $\mathbf{B}_{ur}$ and $\mathbf{B}_{dr}$ in $\mathbf{A}$. Since source patches have already been randomly padded into $\mathbf{S}$, in most cases, $\mathbf{B}_r$ is located in two parts in $\mathbf{S}$, that is the upper region $\mathbf{B}_{ur}$ and the lower region $\mathbf{B}_{dr}$, as shown in Figure 3(d). Therefore, the reconstruction of $\mathbf{B}_r$ is equivalent to the process of finding residual chips $\mathbf{B}_{ur}$ and $\mathbf{B}_{dr}$ in the kernels from $\mathbf{S}$. With that said, we search some cells of $\mathbf{B}_{ur}$ or $\mathbf{B}_{dr}$ in kernels to find matched ones, and extend them to make up $\mathbf{B}'_r$. The recovered $\mathbf{B}'_r$ is same with $\mathbf{B}_r$ in most cases. $\mathbf{B}_l$ is recovered in the same way.

After we reconstruct $\mathbf{B}'_l$ and $\mathcal{B}'$, we are able to get an approximative rank from each OLR and form a rank set $\mathcal{R} = \{r_i | i = 1, 2, ..., N\}$, where $N$ is the amount of ranks. Parenthetically, two types of priority locations for strategy of patch distribution in [12], $\mathrm{L}_1$ and $\mathrm{L}_2$ based resolution, are treated in the same manner. As far as we are aware, a cover synthetic image has all near-zero ranks while a stego synthetic image has much larger ranks. This is why we take the ranks as features to distinguish stego images from cover images. We aggregate the ranks and extract features to implement support vector machine (SVM) training and classification. Four statistics including mean ($\mu_r$), median ($m_r$), variance ($\delta_r$) and kurtosis ($k_r$) of the ranks are chosen for representation. These statistics are exploited to form the feature vector $\mathbf{v}_r$,

$$\mathbf{v}_r = [\mu_r, m_r, \delta_r, k_r]^T, \tag{4}$$

representing a given synthetic image.

If we are unaware of $P_w$, $P_h$ and $P_d$, a traversal process is implemented first to find the three scaling parameters. To gain possible values, Eq. (1) is adjusted to solve an integer programming problem with a

linear equations group:

$$(P_w - P_d) \times T_{pw} + P_d = T_w \tag{5}$$

and

$$(P_h - P_d) \times T_{ph} + P_d = T_h. \tag{6}$$

Since they are underdetermined linear equations, there are several possible solutions. By traversing all the solutions, we have $t$ groups of $\{P_w, P_h, P_d\}$, where the matched group of parameters is homologous to the situation that kernels patitioned from synthetic images are from the original kernels or formed by four kernel fragments in source images that contain no OLRs, as shown in Figure 3(c, d). In comparison, kernels generated from mismatched groups more or less contain OLRs. Such evaluation is effective in discriminating cover images from stego images.

The procedure of steganalysis contains five steps, as shown in Figure 2. In Figure 5(a), the CASO-ReSid pair (black rhombic solid line) shows the effectiveness of proposed ReSid feature. We have provided the source codes for steganalysis on the website[2].

## 3. Security-Enhanced Texture Steganography

In a bid to improve the anti-steganalytic properties of CASO algorithm, we develope a steganographic algorithm utilizing padding technique with redundant areas carrying no data. The four peripheries of synthetic image are broadened with key-specified depths respectively.

Regarding capacity, security and generality, the following insights are given:

- Instead of synthesizing texture images with precalculated sizes under fixed pattern, we are able to set synthesized texture image with arbitrary size and preserve high steganalysis-resistant ability.
- Since CASO uses sliding windows to generate a certain number of candidate patches to carry messages which are restrained by sizes of patches and source image, the maximum embedding payload is $\lfloor \log_2 n_C \rfloor$. By considering that any synthesized patch by two original similar patches are similar to the original ones but not identical, we augment the quantity of candidate patches by synthesizing similar candidate patches to increase the embedding rate.

### *3.1. Synthesized Image with Arbitrary Size*

Additional redundant regions contribute to the security of synthesized texture images, since it is hard for attackers to estimate the size of patches and depth of kernel to further implement steganalysis. Hence, one way to invalid steganalysis is adding redundant regions around the periphery of the synthesized image.

[2]Available: http://home.ustc.edu.cn/∼zh2991/

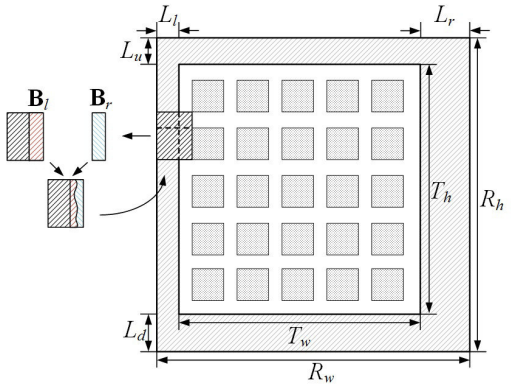



Figure 4: Anticipated synthesized image $\mathbf{R}$ with redundant regions around original synthesized image $\mathbf{S}$. Peripheries are in order generated by MRF-based image quilting method.

Assume the anticipated synthesized texture image $\mathbf{R}$ (slightly larger than $\mathbf{S}$) by $R_w \times R_h$, thus the total accessorial width is $L = (R_w - T_w) + (R_h - T_h)$. To increase the width, two regions are synthesized, as shown in Figure 4, where the redundant widths of left, right, up and down side are $L_l$, $L_r$, $L_u$ and $L_d$ respectively.

Markov Random Field (MRF) is used for texture synthesis. We assume that the probability distribution of values for a region given the values of its spatial neighborhood is independent of the rest of the image. Let $\mathbf{B}_s \in \mathcal{B}$ be the patch to be synthesized. The neighborhood of $\mathbf{B}_s$ is modeled as several rectangles windows around that region. Let $w(\mathbf{B}_s) \subset S$ be the adjacent regions of $\mathbf{B}_s$. To synthesize patch $\mathbf{B}_s$, we first construct an approximation to the conditional probability distribution $P(\mathbf{B}_s|w(\mathbf{B}_s))$ and then sample from it.

Based on our MRF model we assume that $\mathbf{B}_s$ is independent of $S \backslash w(B_s)$ given $\mathbf{B}_s$. The closest match $\mathbf{B}_s^*$ is acquired by

$$\mathbf{B}_s^* = \arg\min_w D(w(\mathbf{B}_s), w), \tag{7}$$

and the patch is padded on $\mathbf{R}$. Likewise, other peripheries are padded until $\mathbf{R}$ is synthesized. Of course, the artifacts of peripheries are fairly mild in that the best matched patches are synthesized.

### *3.2. Capacity Enlargement*

The relative embedding payload $\gamma$ is measured in bit length of message per patch (in bpp), which is related to the performance of steganography. The maximum $\gamma_{max}$ of CASO algorithm depends on the size of source image and depth of kernel region, which is $\gamma_{max} = \lfloor \log_2 n_C \rfloor$. To enlarge the capacity, quantity of candidate patches should be increased without causing certain artifact. Under the assumption that patches with similar complexity share approximate texture structures, one solution is to generate additive candidate patches from the set of existing candidate patches $\mathcal{B}$.

Formally, we cluster the elements of $\mathcal{B}$ to create a new set $\mathcal{B}_s$ with more candidate patches. Let $n_D$ be the number of elements in the subset representing the degree of cluster, and the subset is denoted by $R_{s_i} \subset \mathcal{B}$,

$R_{s_i} = \{B_{i \cdot n_D+1}, B_{i \cdot n_D+2}, ..., B_{(i+1)n_D}\}, i = 1, 2, ..., \left\lfloor \frac{n_C}{n_D} \right\rfloor$. Texture synthesis is implemented between any two elements in each subset $R_{s_i}$, and the total quantity of candidate patches $n_S$ is obtained by

$$n_S = n_C + \left\lfloor \frac{n_C}{n_D} \right\rfloor \binom{n_D}{2}. \tag{8}$$

Thus the maximum embedding rate $\gamma_{max} = \lfloor \log_2 n_S \rfloor$, whose upper bound approaches

$$\begin{aligned} \left\lfloor \lim_{n_D \to n_C} \log_2 n_S \right\rfloor &\leq \left\lfloor \lim_{n_D \to n_C} \log_2 \left( n_C + \frac{n_C}{n_D} \binom{n_D}{2} \right) \right\rfloor \\ &= \log_2 \left[ n_C (n_C + 1) \right] - 1. \end{aligned} \tag{9}$$

To adequately express the message in a patch, we have the following limiting constraint:

$$2^\gamma \leq n_S \leq n_C + \frac{n_C}{n_D} \binom{n_D}{2}. \tag{10}$$

The decision threshold $n_D^*$ is given by:

$$\begin{aligned} n_D^* &= \left\lceil \frac{2^{\gamma+1}}{n_C} - 1 \right\rceil \\ &= \left\lceil \frac{2^{\gamma+1}}{(S_w - P_w + 1)(S_h - P_h + 1)} - 1 \right\rceil. \end{aligned} \tag{11}$$

Thus given an arbitrary embedding rate $\gamma$, degree of cluster $n_D^*$ can be decided.

The total capacity $C$ of the anticipated synthesized texture image $\mathbf{R}$ is

$$C = \left( n_T - \frac{S_w \times S_h}{K_w \times K_h} \right) \gamma, \tag{12}$$

and the enlarged capacity $\Delta C$ compared with that of CASO is

$$\Delta C = \left( n_T - \frac{S_w \times S_h}{K_w \times K_h} \right) \left( \lfloor \log_2 n_S \rfloor - \lfloor \log_2 n_C \rfloor \right), \tag{13}$$

with the overhead of accessorial width $L$ which takes up additional area $\Delta S = R_w \times R_h - T_w \times T_h$.

Notice that the enlarged synthetic image $\mathbf{R}$ has more patches than $\mathbf{S}$, showing that with the same length of embedded message $\mathbf{R}$ has a slightly smaller relative payload $\hat{\gamma}$, which has to be aligned to the original relative payload $\gamma$. The calibrated relative payload is obtained by

$$\hat{\gamma} = \frac{S_w \times S_h}{R_w \times R_h} \cdot \gamma. \tag{14}$$

It is stated in [12] that no significant visual difference exists among pure synthetic image and stego synthetic textures with varying relative payloads. Since the visual artifact around the image periphery is milder than other region with best fit regions padded, we can infer that the anticipated synthesized texture images preserve equal visual quality. We coin a new acronym CASY standing for **Ca**ndidate **Sy**nthesis based texture synthesis steganography.

### 3.3. Security Analysis

In this subsection we discuss the probability of hitting the exact patch size $\{P_w, P_h, P_d\}$. Once these parameters are perceived, with the proposed steganalytic algorithm, CASY algorithm can be broken down.

The eavesdroppers try to crop out redundant width $L_0$ of image periphery to further conduct steganalysis, where the constraint condition lies on that the scope of $L \geq L_0$. The probability of revealing the correct $\mathbf{S}$ from $\mathbf{R}$ is calculated by

$$P_s(L) = \frac{1}{1 + 4 + 4^2 + ... + 4^L} = \frac{3}{4^{L+1} - 1}. \tag{15}$$

As aforementioned, $t$ candidate scaling parameters are concatenated to each $\{P_w, P_h, P_d\}$, causing the ultimate probability of breaking down CASY:

$$P_b(L) = \frac{1}{\sum_{i=1}^{\frac{1}{P_s(L)}} t_i} = \frac{1}{\sum_{i=1}^{\frac{4^{L+1}-1}{3}} t_i}, \tag{16}$$

where $t_i$ is the number of candidate sizes of the $i$-th tentative synthetic image.

We present an example to provide more insight for the security of CASY. Suppose $L = L_0 = 16$, and on average 4 widths increase of pixels on each periphery of synthetic image is created, and mostly smaller than $P_d$, which has an opportunity of $P_s(16) \approx 2 \times 10^{-10}$ to retrieve the exact synthetic texture image. Since $t_i \geq 1$ as an integer varies dissimilarly among sizes, $P_b(16) < P_s(16)$, manifesting that such a brute-force attack on acquiring the accurate $\{P_w, P_h, P_d\}$ is fairly difficult.

In addition, mismatched case: locating the correct position with biased size of $\mathbf{S}$ is experimentally described in Section IV-C, the result of which shows that only the accurately estimated parameter $\{P_w, P_h, P_d\}$ and $L$ provide the most accurate steganalysis results.

## 4. Experiments

The performance of ReSid against Wu *et al.*'s method CASO and the security-enhanced version CASY is validated in Section IV-B and Section IV-C, respectively.

### 4.1. Setups

*1) Database:* All experiments are conducted on Brodatz Textures [14]. First, we use the CASO steganographic method to generate synthetic images. Since texture images are comparatively rare in Brodatz Database, we create some images by cropping and zooming techniques. Finally, we take 10,000 proper images ($128 \times 128$) as source images[2]. Though CASO tests four texture images that are color images, it is uninfluential to the steganalysis if we consider grayscale images, since a preprocessing including transforming color images into grayscale images or adopting one color channel is available. Let us suppose that we wish to design a synthesis and embedment mechanism with a relative payload of $\gamma$ varying from 1 bpp to 13 bpps,

10,000 embedded synthetic images with each relative payload and identical number of synthetic images that are not embedded with messages are generated. In the procedure of texture image steganalysis, we use the same parameters employed in CASO: $T_{pw} = T_{ph} = 488$, $P_d = 8$ and $P_w = P_h = 48$. And the configuration makes $n_C = 6,561$.

Table 1
Detectability in Terms of $\overline{P}_{\mathrm{E}}$ Versus Relative Embedding Payload Size in Bits per Patch (bpp) for CASO and CASY on Texture Database with Three Feature Sets

| Feature | Embedding method | 1 | 3 | 5 | 7 | 9 | 11 | 13 |
|---|---|---|---|---|---|---|---|---|
| SPAM | CASO | .4288 ± .0016 | .4350 ± .0015 | .4669 ± .0017 | .4588 ± .0021 | .4550 ± .0028 | .4509 ± .0015 | / |
| | CASY | .4630 ± .0012 | .4422 ± .0015 | .4720 ± .0022 | .4811 ± .0020 | .4670 ± .0015 | .4637 ± .0015 | .4570 ± .0014 |
| SRM | CASO | .3116 ± .0035 | .3138 ± .0051 | .3105 ± .0067 | .2451 ± .0052 | .1929 ± .0023 | .1413 ± .0020 | / |
| | CASY | .3935 ± .0023 | .3406 ± .0042 | .3741 ± .0161 | .3180 ± .0109 | .2526 ± .0061 | .1908 ± .0051 | .1252 ± .0041 |
| maxSRM | CASO | .2402 ± .0017 | .2009 ± .0042 | .1831 ± .0057 | .1153 ± .0043 | .0739 ± .0012 | .0453 ± .0017 | / |
| | CASY | .3935 ± .0023 | .3406 ± .0042 | .3741 ± .0161 | .3180 ± .0109 | .2526 ± .0061 | .1908 ± .0051 | .1252 ± .0041 |
| ReSid | CASO | .4240 ± .0014 | .2700 ± .0033 | .1840 ± .0031 | .1400 ± .0101 | .1110 ± .0067 | .0500 ± .0032 | / |
| | CASY | .4970 ± .0031 | .4870 ± .0043 | .4640 ± .0021 | .4450 ± .0024 | .4200 ± .0032 | .3880 ± .0019 | .3390 ± .0059 |

*2) Training and Classification:* The texture image steganalysis is evaluated empirically using binary classifiers trained on a given cover source and its stego version embedded with a fixed relative payload. Five-fold cross validation of SVM is employed to conduct training and classification. Each test is repeated 10 times, and results are averaged to evaluate the final performance. Soft-margin SVMs with the Gaussian kernel $k(x, y) = \exp(-\gamma\|x - y\|_2^2)$, $\gamma > 0$ is used. The values of the penalization parameter $C = 5$ and the kernel parameter $\gamma = 0.5$. Our experiments show that Radial Basis Function (RBF) SVM has competitive results, and LIBSVM [15] is utilized here as the classifier for low computing complexity.

We compare results of our features on the generated database with the popular steganalytic features, SPAM [4], SRM [6] and maxSRM [25]. The classifier is implemented using the ensemble [11] with Fisher linear discriminant as the base learner. A number of 5000 randomly selected cover images and their stego counterparts are used for training, while the rest 5000 cover images and their stego counterparts are used for testing. The security is quantified using the ensembles out-of-bag (OOB) error $E_{\mathrm{OOB}}$, which is an unbiased estimate of the minimal total testing error under equal priors [11],

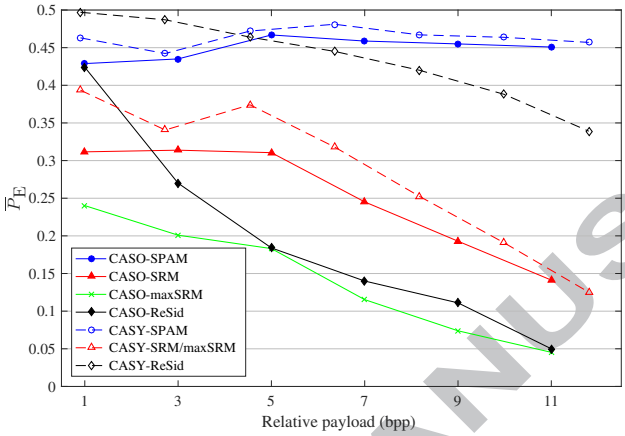
$$P_{\mathrm{E}} = \min_{P_{\mathrm{FA}}} \frac{1}{2}(P_{\mathrm{FA}} + P_{\mathrm{MD}}), \tag{17}$$

where $P_{\mathrm{FA}}$ and $P_{\mathrm{MD}}$ are the false-alarm probability and the missed-detection probability respectively.

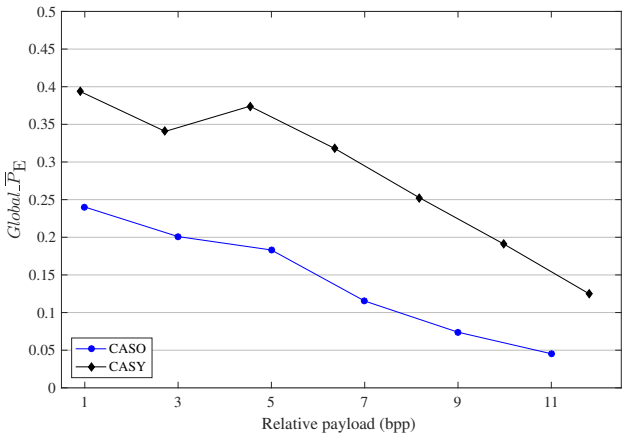
### *4.2. Steganalytic Algorithm Validation on CASO*

We conduct an adaptive search approach to find $\{P_w, P_h, P_d\}$. By solving Eq. (5-6), we get candidate scaling parameters $\mathcal{T} = \{(P_w, P_h, P_d)|(15, 15, 4), (15, 26, 4), ..., (108, 108, 32)\}$ and $\|\mathcal{T}\| = 150$. The result

[2]Texture Database is open to download: http://home.ustc.edu.cn/∼zh2991/




(a)




(b)

Figure 5: (a) Detection error $P_{\mathrm{E}}$ for CASO and CASY schemes when steganalyzing with SPAM [4], SRM [6], maxSRM [25] and ReSid for varying relative payloads. The plot corresponds to the results given in Table I. (b) Global detection error $Global\_P_{\mathrm{E}}$ for CASO and CASY schemes for varying relative payloads.

is effectual with a detection accuracy of greater than 97% finding the matched scaling parameters in both cover and stego images.

Table I shows the average total probability of error $\overline{P}_{\mathrm{E}}$ and its standard deviation for a range of relative payloads for CASO and CASY steganographic schemes described in the previous section. The proposed detector ReSid provides a substantial improvement in detection accuracy over SPAM, SRM and maxSRM feature sets with a linear SVM classifier (see solid lines in Figure 5(a)). For small relative payload with $\gamma = 1$, ReSid is not as effective as SRM, which is most likely because SRM's 34,671-dimentional feature set collects more comprehensive minus difference between cover and stego image than ReSid's 4-dimentional feature set. Throughout the figure, the diversity between cover and stego is imperceptible to SPAM feature, the reason of which might be that the second-order Markov residuals are insensitive to the discrepancy of the two carriers, causing the detection error to be around 45%. Compared with SRM, the performance improvement of ReSid averaged over relative payloads is 5.6%, exhibiting valid and impact feature set over the CASO algorithm.

To make a fair comparison when we consider the scaling parameter as prior knowledge, maxSRM should replace SRM and be considered. We define the map of maxSRM by the following rule: the weights of kenel regions are set by 0 and weights of synthesized regions are set by 1. From Figure 5(a), we can conclude that once steganalyzer has prior of synthesized regions, the detection accuracy increases; our proposed ReSid does not exceed maxSRM but still share near performance when payload is large, but since the dimension of features of ReSid vs. maxSRM is 5 vs. 34,671, the average computation time of maxSRM is 1,455 times much longer than ReSid over varying payloads, showing the superiority of proposed method. As for CASY-maxSRM, map is difficult to acquire and thus the performance of CASY-maxSRM is similar to CASY-SRM. From another perspective, both scaling parameter estimation and ReSid utilize global matching of similar cells to form synthesized region, and to some extent could be set down as a whole.

### *4.3. Security-Enhanced Steganographic Algorithm Validation*

Figure 6 displays visual quality of cover and stego synthetic images generated by CASY algorithm with corresponding source images[3] with a size of $512 \times 512$ and thus $L_0 = 16$. To have a maximum relative payload of $\gamma_{max} = 13$, by Eq. (11), we get the number of clusters $n_D^* = 2$, thus by Eq. (8), number of candidate patches is $n_S = 9841$. Clearly, the enlarged capacity $\Delta C = 128$bits. The advantage of CASY's assignable size might be that since the size information will leak to the eavesdroppers, a normal image size (e.g. $512 \times 512$) is more noteless than patch property specified image size (e.g. $488 \times 488$). No vision disparity exists between cover and stego image.

[3]The four demonstration image (a∼d) are randomly selected from Texture Base we collected, which are '1.bmp', '4.bmp', '3110.bmp' and '8762.bmp' respectively.

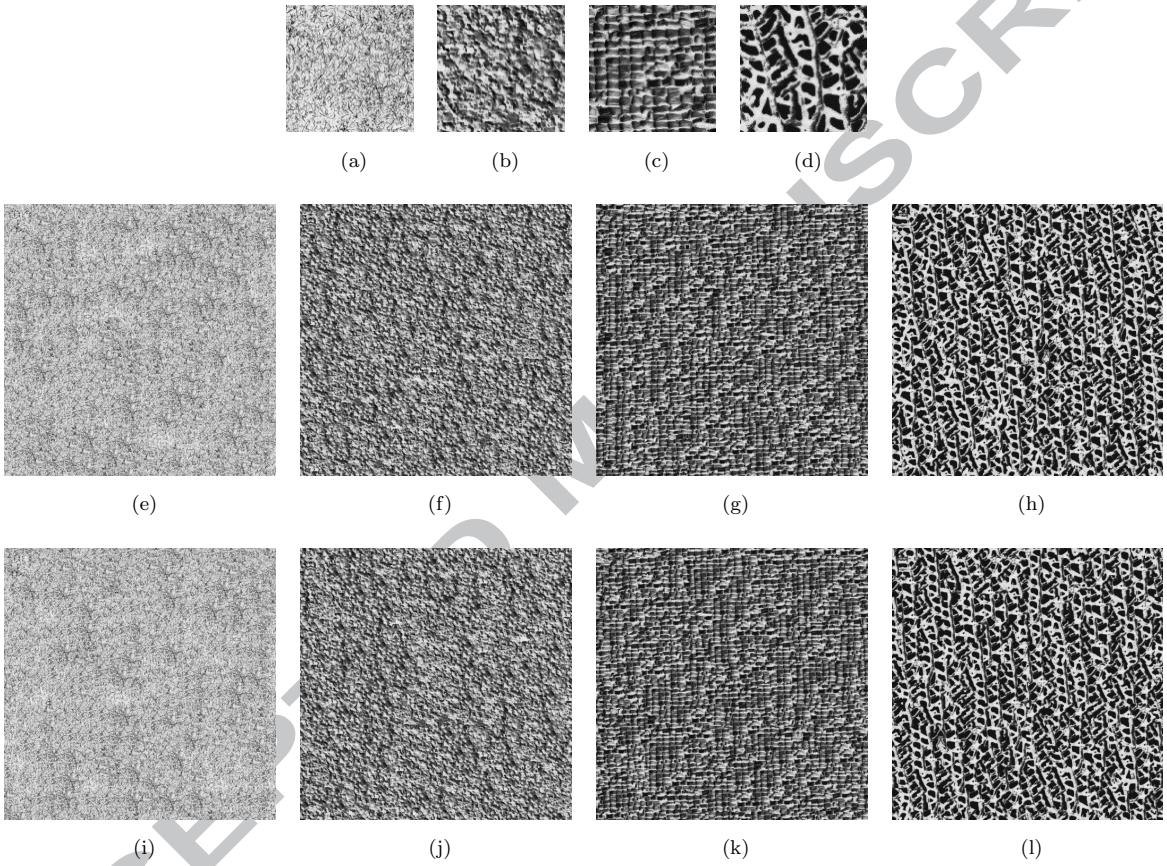



Figure 6: Source texture images and corresponding synthetic texture images. (a)∼(d) are the source images. (e)∼(h) are the synthesized texture images containing no secret messages. (i)∼(l) are the synthesized texture images containing secret messages, relative payloads of which are 5 bpps.

Steganalysis is executed on CASO and CASY methods, which are shown in Figure 5(a). To make a fair comparison, we use the calibrated relative payload $\hat{\gamma}$ given in Eq. (14). To align to the original $\gamma$, $\hat{\gamma} = 0.908\gamma$. The CASY based scheme improves the level of security under SPAM, SRM and maxSRM, contributing to the manipulation that adjacent similar patches are synthesized. While ReSid cannot be directly utilized in steganalysis on CASY, a preprocessing with rough estimations of size of $\mathbf{S}$ are implemented before using ReSid. We take $T_w = T_h = 489$ to conduct steganalysis, and the selection region is located on the center of $\mathbf{R}$, which shows the steganalysis results with a little deviation of estimation of parameters $\{P_w, P_h, P_d\}$ and $L$. Values of parameter $\{P_w, P_h, P_d\}$ are thus estimated: $\{45, 45, 8\}$. The results show that ReSid is capable of detecting CASO than SPAM and SRM features, and a biased estimation of $T_w$ and $T_h$ is still unable to conduct effective steganalysis on CASY than SRM feature.

Note that one steganographic method is broken as long as there exists one steganalytic algorithm that can detect it with a high accuracy rate. Therefore, we introduce another measurement $Global\_P_{\mathrm{E}}$ to depict the comprehensive undetectable ability of the steganographic method [24]:

$$Global\_P_{\mathrm{E}} = \min_{i \in \mathbf{F}} P_{\mathrm{E}}^{i}, \tag{18}$$

where $\mathbf{F}$ represents the set of used steganalysis algorithms. $P_{\mathrm{E}}^{i}$ is the value of $P_{\mathrm{E}}$ under the attack of the $i$-th steganalysis algorithm.

The comprehensive security performance on resisting SPAM, SRM, maxSRM and ReSid is shown in Figure 5(b). It can be seen that the proposed CASY outperforms CASO method with an average improvement of more than 10%. Apart from SPAM, Steganalysis algorithm SRM and ReSid attack the steganographic methods based on two different respects. SRM is designed using the statistical characteristics change in local regions and ReSid utilizes the optimality between adjacent synthesized patches. In our proposed CASY method, we not only invalidate ReSid feature with obscure $\{P_w, P_h, P_d\}$, but also preserve the optimality of synthesized patches with increased similar patches and suppress the associated prediction error during data embedding. Meanwhile, the inability of acquiring map makes maxSRM degenerate to SRM. Therefore, the proposed method can obtain a better comprehensive security performance than CASO method.

## 5. Conclusion

As demonstrated by the experimental results, the developed steganalysis (ReSid) is able to detect the parameters of patches by Wu *et al.*'s [12] algorithm (CASO) with an accuracy of 97%, and with a computational time 1,455 times faster than maxSRM detector under some degradation of detection error. While the proposed steganalytic algorithm was specifically designed to target Wu *et al.*'s algorithm, the main idea could be applied on several other algorithms that embed data with patch synthesis based steganography since a preprocessing of patch parameter estimation is proposed with a high accuracy.

The proposed steganographic algorithm (CASY) is based on a random padding carrying no message around the periphery of the synthesized image to invalidate the parameter estimation of patches. To enhance security and improve capacity, additional candidate patches are generated through synthesizing similar original candidate patches. The experimental results demonstrate that it outperforms Wu *et al.*'s algorithm in terms of security and capacity.

In the future, we will work to apply the proposed steganography method to texture synthesis related applications such as online games, 3D roads, and virtual reality.

## References


[1] A. D. Ker, P. Bas, R. Böhme, R. Cogranne, S. Craver, T. Filler, J. Fridrich, and T. Pevný, "Moving steganography and steganalysis from the laboratory into the real world," in *Proceedings of the first ACM workshop on Information hiding and multimedia security*, pp. 45–58, ACM, 2013.

[2] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 920–935, 2011.

[3] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *International Workshop on Information Hiding*, pp. 161–177, Springer, 2010.

[4] T. Pevný, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 215–224, 2010.

[5] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 234–239, IEEE, 2012.

[6] J. Fridrich and J. Kodovský, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[7] V. Holub and J. Fridrich, "Digital image steganography using universal distortion," in *Proceedings of the first ACM workshop on Information hiding and multimedia security*, pp. 59–68, ACM, 2013.

[8] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 4206–4210, IEEE, 2014.

[9] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 221–234, 2016.

[10] J. Kodovský, T. Pevný, and J. Fridrich, "Modern steganalysis can detect YASS," in *IS&T/SPIE Electronic Imaging*, pp. 754102–754102, International Society for Optics and Photonics, 2010.

[11] J. Kodovský, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 432–444, 2012.

[12] K.-C. Wu and C.-M. Wang, "Steganography using reversible texture synthesis," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 130–139, 2015.

[13] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 341–346, ACM, 2001.

[14] "Brodatz Texture Database." [Online], 1997. http://multibandtexture.recherche.usherbrooke.ca/original_brodatz.html.

[15] "LIBLINEAR-A Library for Large Linear Classification." [Online], 2015. http://www.csie.ntu.edu.tw/%7Ecjlin/liblinear.

[16] S. Lyu and H. Farid, "Detecting hidden messages using higher-order statistics and support vector machines," in *International Workshop on Information Hiding*, pp. 340–354, Springer, 2002.

[17] J. Fridrich, M. Goljan, and R. Du, "Detecting LSB steganography in color, and gray-scale images," *IEEE multimedia*, vol. 8, no. 4, pp. 22–28, 2001.

[18] A. D. Ker, "Steganalysis of LSB matching in grayscale images," *IEEE signal processing letters*, vol. 12, no. 6, pp. 441–444, 2005.

[19] H. Otori and S. Kuriyama, "Data-embeddable texture synthesis," in *International Symposium on Smart Graphics*, pp. 146–157, Springer, 2007.

[20] H. Otori and S. Kuriyama, "Texture synthesis for mobile data communications," *IEEE Computer graphics and applications*, vol. 29, no. 6, pp. 74–81, 2009.

[21] Z. Qian, H. Zhou, W. Zhang, and X. Zhang, "Robust steganography using texture synthesis," in *Proceeding of the Twelfth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Nov., 21-23, 2016, Kaohsiung, Taiwan, Volume 1*, pp. 25–33, Springer, 2017.

[22] H. Zhou, K. Chen, W. Zhang, and N. Yu, "Comments on "steganography using reversible texture synthesis"," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1623–1625, 2017.

[23] W. Zhou, W. Zhang, and N. Yu, "A new rule for cost reassignment in adaptive steganography," *IEEE Transactions on Information Forensics and Security*, 2017.

[24] Y. Yao, W. Zhang, N. Yu, and X. Zhao, "Defining embedding distortion for motion vector-based video steganography," *Multimedia Tools and Applications*, vol. 74, no. 24, pp. 11163–11186, 2015.

[25] T. Denemark, V. Sedighi, V. Holub, R. Cogranne, and J. Fridrich, "Selection-channel-aware rich model for steganalysis of digital images," *Proc. of 6th IEEE International Workshop on Information Forensics and Security*, Atlanta, GA, USA, Dec. 3-5, 2014.

# Highlights

- Build an effective and efficient steganography detector for patch synthesis based steganography and thus improve the steganalysis performance to a large extent.
- Improve the maximum embedding capacity of steganography compared with previous art.
- Improve the security of steganography against state-of-the-art methods and steganalytic attack we developed.