Soft Computing
https://doi.org/10.1007/s00500-018-3151-8

METHODOLOGIES AND APPLICATION

Coverless image steganography using partial-duplicate image retrieval

Zhili Zhou[1] · Yan Mu[1] · Q. M. Jonathan Wu[2]

Abstract
Most of the existing image steganographic approaches embed the secret information imperceptibly into a cover image by slightly modifying its content. However, the modification traces will cause some distortion in the stego-image, especially when embedding color image data that usually contain thousands of bits, which makes successful steganalysis possible. In this paper, we propose a novel coverless steganographic approach without any modification for transmitting secret color image. In our approach, instead of modifying a cover image to generate the stego-image, steganography is realized by using a set of proper partial duplicates of a given secret image as stego-images, which are retrieved from a natural image database. More specifically, after dividing each database image into a number of non-overlapping patches and indexing those images based on the features extracted from these patches, we search for the partial duplicates of the secret image in the database to obtain the stego-images, each of which shares one or several visually similar patches with the secret image. At the receiver end, by using the patches of the stego-images, our approach can approximately recover the secret image. Since the stego-images are natural ones without any modification traces, our approach can resist all of the existing steganalysis tools. Experimental results and analysis prove that our approach not only has strong resistance to steganalysis, but also has desirable security and high hiding capability.

Keywords Coverless steganographic approach · Partial-duplicate image retrieval · Stego-image · Capacity · Security

1 Introduction

Steganography is the art of communicating secret information in a hidden manner. Generally, it hides secret information imperceptibly into an appropriate host medium such as digital image, audio, and video files so that the presence of hidden information cannot be diagnosed. Thus, different from cryptography, in which the secret communication is evident, steganography can conceal the very existence of the secret communication itself (Cheddad et al. 2010).

In the last two decades, many image steganographic approaches (Wu et al. 2005; Mielikainen 2006; Johnson and Jajodia 1998; Li et al. 2009; Li and Wang 2007; Chen 2007; Mckeon 2007; Chang et al. 2009; Luo et al. 2010; Kawaguchi 2005; Hioki 2002) have been proposed with the wide use of digital images (Liao et al. 2017a). Among those approaches, most of them select an innocuous image as a cover and then embed the secret data into the cover by modifying its content to generate the stego-image. However, the modification traces will cause some image distortion in the stego-image, especially when embedding the secret data with relatively large data length, such as color image data that usually contains thousands of bits. Consequently, the existence of image distortion in the stego-image makes successful steganalysis possible.

Therefore, if we adopt a hiding manner in which the secret information can be hidden without any modification, it will be effective to resist all of the existing steganalysis tools. How to hide the secret information without any modifica-

Communicated by V. Loia.

✉ Q. M. Jonathan Wu
jwu@uwindsor.ca

Zhili Zhou
zhou_zhili@163.com

Yan Mu
muyan_my@163.com

[1] School of Computer and Software and Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing 210044, China

[2] Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada

Published online: 27 March 2018

tion? It is an exciting and challenging task. As we know, any image contains a lot of information. It is possible that the secret information needed to be hidden is already contained in the image. Fridrich and Kodovsky (2012) introduce a novel image steganography-based cover selection. This method selects images from the database according to the secret information to represent the secret information, so that the secret information can be hidden in the images. In our previous work, a novel steganographic approach based on robust hashing algorithm is proposed (Zhou et al. 2015), and it is based on the idea of coverless image steganography. Coverless (Cao et al. 2018; Xia and Li 2017) image steganography is a new concept to realize the information hiding. It does not need to designate and modify a cover image to hide the secret information. Instead, the hiding process is implemented by finding an image or text that already contains the secret information (Zhou et al. 2015, 2017a). The method is to build mapping relationships between the hash sequences and the secret messages. To the best of our knowledge, two other methods (Zhou et al. 2016a; Yuan et al. 2016) are also based on the idea of information mapping for the coverless image steganography. One of them is to build the information mapping between image visual words and the secret information (Zhou et al. 2016a). Another one is to build the information mapping between the SIFT features of images and the secret information (Yuan et al. 2016). All the above methods employ the information mapping instead of modifying the cover image to generate the stego-image. Consequently, if we hide n-bit secret information into an image, it needs 2^n natural images to transmit secret data. However, if there is a secret image which contains n-bit information that needs to be hidden, about 2^n images are required to represent the secret information. Thus, the number of images increases exponentially with the length of the secret data, which makes those approaches impractical.

Nevertheless, for a given secret image, it is possible that some visually similar patches may exist between the secret image and its partial duplicates, as shown in Fig. 1a. If we can find a set of partial duplicates of the secret image in a natural image dataset, we can use these partial duplicates as stego-images to represent and hide the secret image. Note that, since the partial duplicates are retrieved from natural image dataset, we can realize the information hiding without any modifications. At the receiver end, the similar regions can be cropped from the stego-images and spliced together to approximately recover the secret image. Figure 1b shows an example of the secret image recovered from its partial duplicates.

Therefore, in this paper, we propose a novel steganographic approach based on partial-duplicate image retrieval. In our approach, we divide the images from the database into a number of image patches (Liao et al. 2017b), and then, these images are indexed by using the features extracted from the image patches. To hide the secret image, the secret image will be also divided into several image patches, and then, we search for the partial duplicates of the secret image based on the image patch similarity. At the receiver end, our approach can approximately recover the secret image from these partial duplicates.

As a summary, the proposed approach has three main advantages.

1. Strong resistance to steganalysis. It has strong resistance to the existing steganalysis tools, since the retrieved partial duplicates that are natural images are used as stego-images to transmit the secret data.
2. Desirable security. Instead of using only one image, we use a set of natural images to hide the secret image. On the other hand, we also employ a key to control and decide which part of a natural image is used for hiding secret information. Thus, it is also hard to recover the secret image by the crackers.
3. High capability. A given secret image and its partial duplicates may share one or several similar patches. Thus, we can hide one or several secret patches, which consist of thousands of bits, into each partial duplicate.

The reminder of this paper is organized as follows. In Sect. 2, we review the related work, including the classical image steganography methods, the steganography methods using texture synthesis, and the existing coverless steganography methods. In Sect. 3, we detail our approach, which has three main components: the construction of index file, hiding the secret image, and recovering the secret image. In Sect. 4, the experimental results and analysis are described in detail. Section 5 gives our conclusions and future work.

2 Related work

In the past few decades, many image steganography methods have been proposed. Most of these methods assign an appropriate cover image and embed some secret information into it to generate the stego-image. The existing steganography methods can be divided into the following three classes: spatial domain, frequency domain, and adaptive methods. LSB replacement (Wu et al. 2005), LSB matching (Mielikainen 2006), color palette (Johnson and Jajodia 1998), and histogram-based methods (Li et al. 2009) belong to the typical spatial domain methods. Typical frequency domain methods include quantization table (Li and Wang 2007), discrete wavelet transform-based embedding (Chen 2007; Xiong et al. 2017; Wang et al. 2017), and discrete Fourier transform (Mckeon 2007). Generally, frequency domain methods are proposed to address the robustness problem to the image attacks (Chen et al. 2017a). The locally adaptive coding-based (Chang et al. 2009; Yin et al.

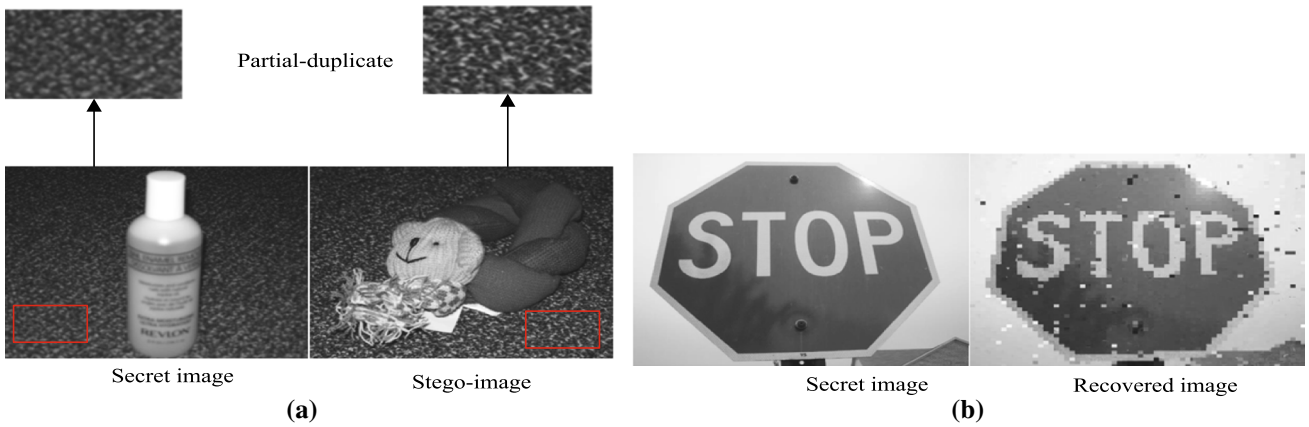

Fig. 1 **a** A natural image shares one similar patch with the secret image. **b** The original secret image and the secret image recovered by our approach

2017a, b), edge-based (Luo et al. 2010), and Bit Pane Complexity Segmentation-based data embedding (Kawaguchi 2005; Hioki 2002) are the typical adaptive steganography. However, all of these methods have a common problem that they modify the cover image to realize the embedding process. Consequently, the existing steganalysis tools can easily uncover that whether there are modification traces left in the cover image.

Recently, some approaches are proposed for steganography using texture synthesis (Wu and Wang 2014). As illustrated in (Wu and Wang 2015), the original aim of texture synthesis is to resample a small source texture image to synthesize a larger texture image. These approaches use the basic texture image to synthesis a new and larger texture image. During the texture synthesis process, these methods can conceal the secret information by selecting different patches of basic texture image. Moreover, the image-quilting algorithm (Efros and Freeman 2001) is adopted to reduce the visual artifact on the overlapped area of adjacent source patches. They show the high capacity as well as good ability to resist the detection of steganalysis tools. Unfortunately, the up-to-date comments given in (Wu and Wang 2014) demonstrate the steganographic approaches using texture synthesis are not secure enough. The attackers can guess the source texture image by using the quilting traces, and thus, they can recover the secret data.

Steganography methods by cover selection (Fridrich and Kodovsky 2012; Sun et al. 2017; Chen et al. 2017b) are to choose an image from a fixed database of images to hide the secret information. The motivation of these methods is to transform a given image into several binary bits by hash functions to realize information hiding. For the binary bits of the given secret information, it selects image of which binary bits are equal to the secret binary bits to hide the secret information. At the receiver end, we can easily extract the information from the stego-image. Similarly, our previous work (Zhou et al. 2015, 2016a; Yuan et al. 2016) is to construct the relationships between the cover image and the secret information by using the local features extracted from the cover image. For the given secret information, they divide the secret information into several information units. Then, they find the image of which feature is equal to one of the unit. After that, we can obtain several images as stego-images, the number of which is the same to the number of information units. These methods embed information without any modification on the cover image. However, they usually have low hiding capacity, which makes them impractical in real-word applications.

Recently, bag-of-visual-words (BOW) (Sivic and Zisserman 2003) is proposed for the tasks of large-scale near-duplicate image retrieval (Sivic and Zisserman 2012; Zhou et al. 2016b). To the best of our knowledge, the main idea of BOW model is to quantize local features extracted from images into visual words by k-means algorithm. Then, images are indexed with inverted structure for image retrieval. However, the main problem of this model is that the retrieval efficiency is limited, when we search for an image in a very large image database. Some new image retrieval methods such as hierarchical k-means-based BOW model (Kieu et al. 2016) are proposed to improve the traditional BOW model. The efficient near-duplicate image retrieval methods (Zhou et al. 2014, 2016c, 2017b) can rapidly retrieve the near-duplicate images from the image database. Also, we find that an image contains many image patches, which may be similar to the patches of the other images. This gives us some inspiration in coverless image steganography. More specifically, for a given secret image, we can search for a set of its partial duplicates to represent and hide the secret image to realize the information hiding.

Therefore, we propose a novel coverless steganographic approach based on partial-duplicate image retrieval for communicating secret color image. Since the partial duplicates can be natural images, our proposed approach embeds the secret information without any modification on the cover

image. Therefore, it is hard to uncover the existence of the hidden information by the existing steganalysis tools. Moreover, since one partial-duplicate image usually shares one or several similar patches with the secret image, the capacity of the proposed approach can be relatively high. In addition, we can employ a key to control and decide which part of a natural image is used for hiding secret information to achieve good security.

3 The proposed steganography approach

In this section, we will illustrate the secret image hiding and extracting procedures. Figure 2 shows the flow chart of our approach.

In our approach, we collect a large number of images to construct a large-scale database from networks. Then, we divide each of those images into a number of non-overlapping patches and compute a label for each patch of a given image by using robust hashing algorithm.The label is used as the location information. The location information of the image indicates which patch of the image is used for hiding secret information. Note that the location information is shared as a secret key between sender side and receiver side. We extract the feature from each image patch and use the hierarchical BOW to build an inverted index structure.

To conceal the secret image, we first divide the image into several patches with the same size. Then, for each patch, the partial-duplicate image that contains the similar patches with the secret image is retrieved by using the inverted index files. Afterward, a number of partial-duplicate images are obtained, which can be considered as stego-images. Then, those stego-images are transmitted to the receiver.

At the receiver end, we also use the location information to extract those patches from the stego-images. Because the secret image and its partial duplicates share one or several similar patches, we can stitch those patches together to recover the secret image.

As a summary, the main parts of this approach include construction of inverted index files, partial-duplicate image retrieval by the inverted index files, and recovering the secret image in the receiver end.

3.1 The process of constructing inverted index files of image database

To facilitate the partial-duplicate search in our steganography approach, we extract local features from non-overlapping image patches and index the patches by their features. There are three important steps, which are detailed as follows.

Dividing the image into several image patches The basic unit used in our approach is an image patch, of which size is defined by users. In our approach, the size of image patch should keep consistent in the process of information hiding and extracting. Figure 3 illustrates the image division process of our approach. Denote the size of a given image as $I_w \times I_h$, and one of its image patch is $P_w \times P_h$. Also, let $\|PB\|$ represent the number of image patches, and the i-th image patches are denoted by pb_i, i.e., $PB = \{pb_i \,|i = 0 \text{ to } \|PB\| - 1\}$.

As an example, given an image with the size $I_w \times I_h = 640 \times 480$, if we set the size of image patch $P_w \times P_h = 64 \times 48$, we can generate $\|PB\| = 100$ patches. The number of patches can be calculated using (1).

$$\|PB\| = \frac{I_w}{P_w} \times \frac{I_h}{P_h} \tag{1}$$

Feature extraction for each image patch If we find the partial-duplicate images that share quite similar region with the secret image, we can approximately recover the secret image from those partial-duplicates and the recovered image may quite visually similar to the original secret image. Therefore, the region of image needs to be described sufficiently by capturing the image characteristics. To this end, we extract the classic image feature FK, i.e., gray histogram to capture the image characteristics.

FK represents the final feature and fk is the feature from each region. As we know, the gray histogram captures the gray-level information of images. However, the gray histogram is a global feature, which lacks the spatial information. In order to encode the spatial information, we further divide each patch into $m \times n$ regions. In each region, we extract the gray histogram, respectively. Then, we concatenate all of the features to form the final feature. Figure 4 illustrates the process of extracting the final feature and set $m \times n = 4 \times 4$. According to this, we assume that the number of dimensions of gray histogram in each region is k, and each patch contains $m \times n$ regions. We can get the final feature by using (2).

$$FK = \left\{fk^1, fk^2, fk^3, \ldots, fk^{m \times n}\right\} \tag{2}$$

Construction of inverted index files To improve the security of the approach, we should use different patches in the partial-duplicate images to hide the secret information. To this end, we not only need the image patch, but also define the label of the image patch, indicating which part of the partial-duplicate images will be used for information hiding. Therefore, the inverted index structure file is constructed to include two parts, the location information of the image patch and visual words. We use hierarchical BOW quantization to build the inverted index file, which is shown in Fig. 5.

To use the location of image patch, we employ the robust hashing algorithm to generate a hash sequence to locate the each image patch denoted as location information. The main

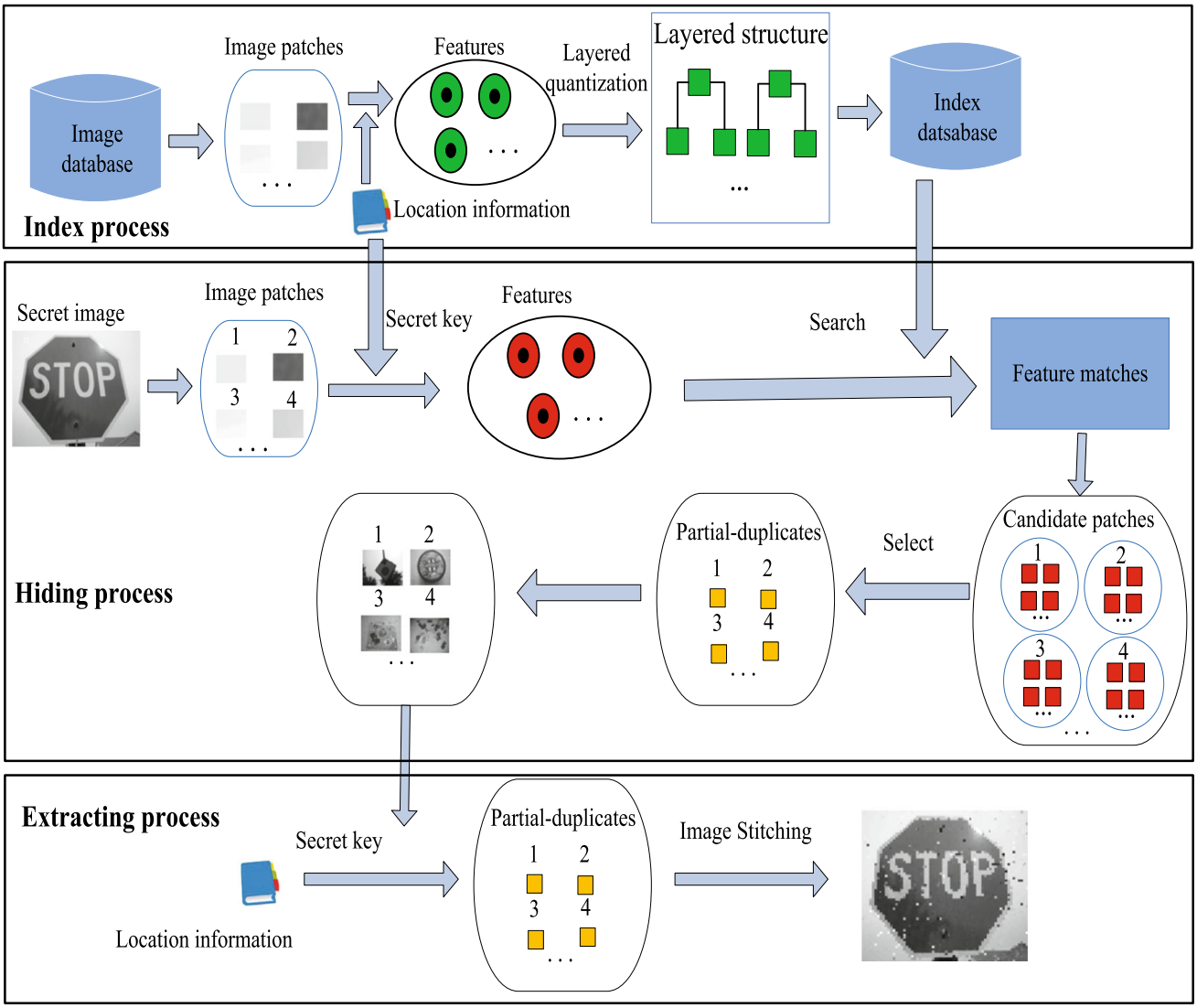

Fig. 2 The flowchart of our approach for secret image communication

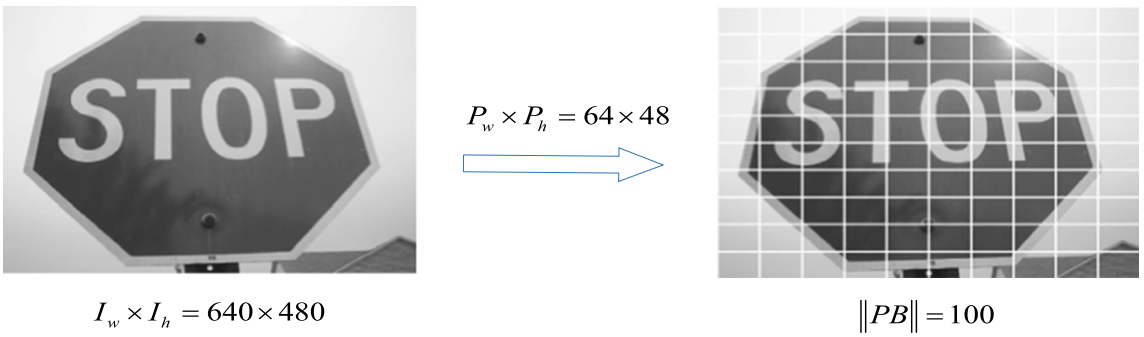

Fig. 3 The way of dividing the image

idea of the robust hashing algorithm is divided into three steps. Firstly, the complete image would be transformed to the gray-level image and divide the image into $m \times n$ secret image patches, denoted as $\{b_{11}, b_{12}, \ldots, bmn\}$. Second step is to compute mean pixel of each image patch and then get $m \times n$ values $\{V(b_{11}), V(b_{12}), \ldots, V(b_{mn})\}$. Third step is to concatenate the values in a zigzag order to a vector, which is denoted as $\{V_1, V_2, \ldots, V_{m*n}\}$. Each value V_i in

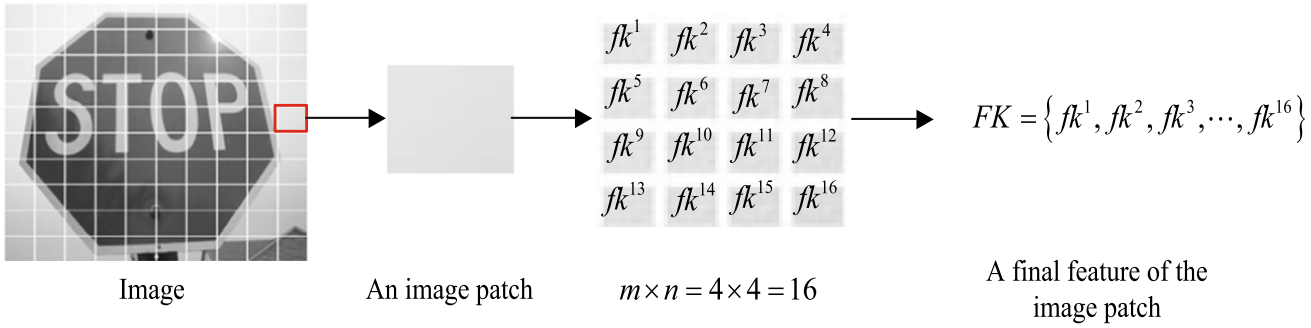

Fig. 4 The process of extracting the final feature of an image patch

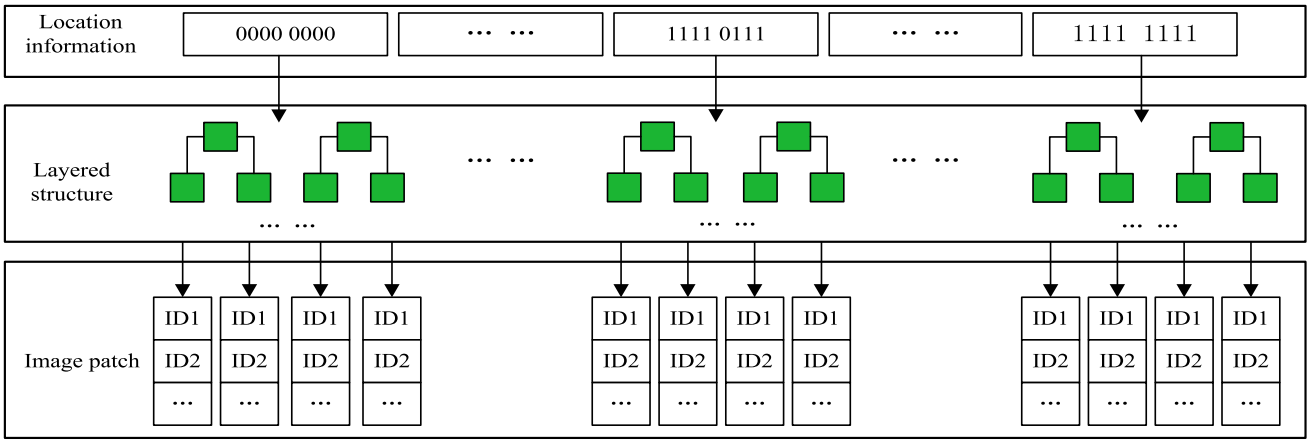

Fig. 5 The inverted index structure using hierarchical BOW quantization

the vector would be compared to its adjacent V_{i+1} by using (3) to generate a hash sequence of the image denoted as $\{v_1, v_2, \ldots, v_{m*n-1}\}$. The process of hashing algorithm is shown in Fig. 6.

$$\begin{cases} v_i = 1, & \text{if } V_i \geq V_{i+1} \\ v_i = 0, & \text{otherwise} \end{cases}, \quad \text{where } 1 \leq i \leq n-1 \tag{3}$$

After obtaining the location information, we will classify image patches according to the location of each image patch. If the length of location information is m, we will get 2^m categories of location information. Due to this, a series of the location information SL can be obtained. Let $SL = \{sl_i \,|i = 0 \text{ to } i = 2^m - 1\}$ represents the set of location information.

Then, we will use the BOW quantization for each patch group. There are several steps to implement the BOW quantization. The process of implementing the BOW quantization is described as follows. First, we extract the gray histogram from each image patch and put these features together in a dataset. Second, we employ the k-means algorithm to cluster the dataset to obtain k cluster centers. After performing this operation, we will get sub-datasets. The third is to repeat the same clustering algorithm on those sub-datasets. We can define L to represent the number of layers. By clustering each sub-dataset, we will get a layer. If the number of layers is larger than L, which is defined previously, we will terminate the cluster operation. Then, a hierarchical BOW quantization structure is constructed. Finally, we will index all the image patches from the image in database by using the hierarchical BOW algorithm.

3.2 Secret image hiding by partial-duplicate image search

In this part, the process of hiding the secret image by partial-duplicate image search is divided into four steps, and the pseudocode is shown as follows.

1. For a given secret image needed to be hidden, similar to the process of database image, we can divide it into a set of patches. Also, we adapt the same way to divide the secret image using (1).
2. To improve the security of our approach, the location information is employed for hiding the secret message. Therefore, we define a secret key, which is shared between the sender and receiver beforehand. By the given secret key, we generate a set of label values to indi-

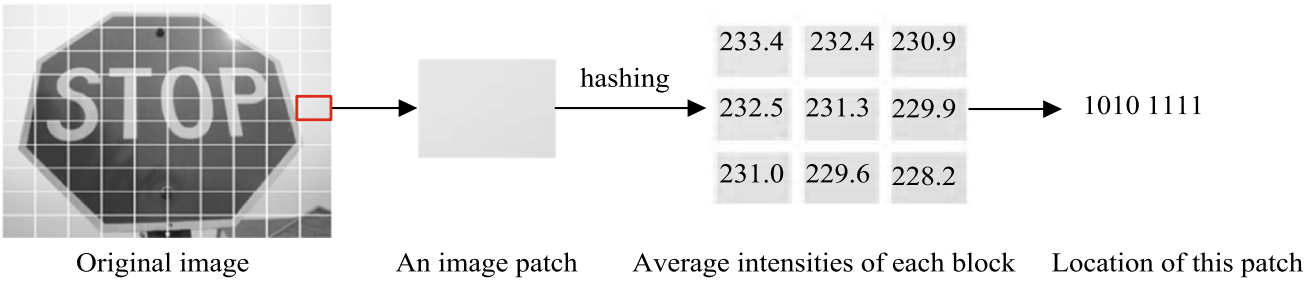

Fig. 6 The process of generating the location by the robust hashing algorithm

cate which part of a partial-duplicate image is used to hide secret information. More specifically, by defining the given secret key K and original location information SL, we get a new set of location information $\overline{SL}$ denoted as $\overline{SL} = \{sl_i \,|i = 0 \text{ to } i = n\}$. Then, according to the new set of label, we can get the duplicate image patches and their corresponding partial-duplicate images by looking up the constructed inverted index file. In some case, the number of image patches may be larger than the number of the locations. To solve this problem, we may plan to reuse the location information $\overline{SL}$. According to this, the number of cycles T is defined using (4). After doing this, the final location information is composed of T new location information $\overline{SL}$ denoted as $\widetilde{SL} = \{sl_i \,|i = 0 \text{ to } i = T \times n - 1\}$.

$$T = \left\lceil \frac{\|PB\|}{n} \right\rceil \tag{4}$$

3. After the above steps, we can obtain many partial duplicates from the image database using the index files. How to choose an appropriate partial duplicate from those candidate duplicates is an important issue. In our approach, for each label value, we may get a set of partial duplicates that are partially similar to the secret image patch. We can select one from those candidates randomly to meet our needs. However, in this way, the visual quality of the recovered secret image is not the best. Hence, we will compute the Euclidean distance between the feature of the secret image patch and candidates using (5). Then, we will select the final partial-duplicate image to represent the secret image patch, which is the smallest distance between the secret image patch and candidates. Let $DP = \{dp_i \,|i = 0 \text{ to } i = \|PB\| - 1\}$ represent the set of partial duplicates.

$$D = \sqrt{\sum_{i=0}^{FK} (f_s^i - f_c^i)} \tag{5}$$

4. We need to send a complete image to receiver, rather than an image patch. Therefore, the partial duplicates should be replaced by their original image. Let $C = \{c_i \,|i = 0 \text{ to } i = \|PB\| - 1\}$ represent the original images, which contain each final partial duplicate.

Algorithm 1 The secret image hiding algorithm

Input: input the secret image, secret key K, and location information SL
Output: output the stego-images C
1: Divide secret image into serval image patches PB;
2: Obtain a new series location information $\widetilde{SL}$;
3: **for** $i \leftarrow 0$ to $\|PB\| - 1$ **do**
4: Retrieve the image patch dp_i under location sl_i using index files and pb_i;
5: Replace dp_i with c_i;
6: Put c_i into the set C;
7: All images in set C are stego-images

3.3 Secret image extracting procedure

The secret image extracting for the receiver side involves obtaining the location information and extracting the secret image concealed in the stego-images.

Given a secret key held in the receiver side, the same location information as the embedding procedure can be obtained. The next step is the secret image recovery, which contains following sub-steps. The first sub-step is that receiver side obtains the each partial duplicate from the original image using the location information. The second is to give a blank area which size is the same to the secret image. Finally, receiver side places these duplicates in the blank area one by one to generate an image, which is similar to the secret image. The pseudocode is shown as follows.

3.4 The computational cost of the proposed method

In the proposed method, the computational cost is mainly the time complexity. And it means that how much time we retrieve partial duplicates from the index files. When we divide the secret image into n image patches, the layer of

Algorithm 2 The secret image extracting algorithm

Input: the stego-images C, secret key K, and location information SL
Output: output the recovery image

1: Obtain a new series location information which is same to embedding procedure;
2: **for** $i \leftarrow 0$ to $\|PB\| - 1$ **do**
3: Extract dp_i from c_i using sl_i;
4: Put into the set ;
5: Obtain DP;
6: Stitching dp_i one by one;
7: Obtain the recovery image

Table 1 Embedding capacity of our approach

$P_w \times P_h$	IC (bit)
40×30	9600
20×15	2400
8×6	384
4×3	96

the structure is l, the number of clusters is k, the length of feature is f, and the time complexity is represented by $T(n) = n * l * k * f$. Meanwhile, the symbols l, k, f are the constant in the method. Therefore, the time complexity of the proposed method is defined as $\Theta(n)$.

4 Experimental results and analysis

4.1 Results of the embedding capacity

We conduct our experiments on a standard computer with an E5-2650 2.60 GHz CPU and 24 GB memory. We adopt three secret images for the results of our collection, and we resize each secret image 640×480. In our approach, we use the gray image as the secret image, and the capacity is defined as the number of bits hidden in one stego-image. Therefore, we assume that a secret image size is $I_w \times I_h$, and the secret image can be hidden successfully by using m stego-images. The capacity IC is calculated using (6).

$$\mathrm{IC} = \frac{I_w \times I_h \times 8}{m} \tag{6}$$

It is necessary to point out that the image patches are with lager size, the higher capacity we will get. Due to this, we set the size of image patch as $P_w \times P_h = 40 \times 30$, $P_w \times P_h = 20 \times 15$, $P_w \times P_h = 8 \times 6$ and $P_w \times P_h = 4 \times 3$. Table 1 presents the capacity of our approach upon different image patch sizes.

We compare our embedding capacity with three existing coverless image steganography methods. Their approaches are coverless image steganography without embedding (Zhou et al. 2015), denoted as CIS-PDVR, coverless image steganography method proposed in reference (Zhou et al. 2016a), denoted as CIS-BOW, and coverless image steganography based on SIFT and BOF, denoted as CIS-BOF (Yuan et al. 2016). In their algorithms, a binary sequence can be concealed in one stego-image. Table 2 shows the comparison of the hiding capacity between our methods and the other methods. As shown in Table 2, our capacity is higher than those of the other methods.

Table 2 The capacity comparison

Method	IC (bit)
Our proposed method	384
CIS-PDVR	8
CIS-BOF	8
CIS-BOW	16

4.2 Recovery image quality

The visual quality of the recovered secret image is the most important in the process of evaluation. Thus, a proper image quality measuring method is required to accurately measure the similarity. The SSIM (Structure Similarity) index (Wang et al. 2004) is an image quality evaluation method that the evaluate result seems more consistent with the subjective sensation of people. Therefore, the SSIM index is used to quantify the similarity between the secret image and recovery image.

The SSIM index is a full-reference image quality measuring method that it can compare the two images with their illuminance, contrast, and structure to calculate the similarity between these two images. The SSIM index is in the range of $[-1, 1]$, and when it equals to one, the two images are identical (Wu and Wang 2014). Figure 7 shows the SSIM index of 50 secret images which are selected from the image database. The x-axis is the number of the secret image, and y-axis is the SSIM index corresponding to each secret image and recovery image. From the figure, we can find that the SSIM indexes are mostly between 0.47 and 0.74. For getting this result, the average number of image patches in each category is about 8 millions.

During the experiment, we find that the number of image patches and the size of image database also affect the quality of recovery image. When we divide the secret image into more image patches, the quality of recovery image is the same to the secret image. Table 5 presents the quality of each secret image under different amounts of image patches. However, if there are more image patches, the capacity is 8-bit, which is very low. Therefore, we need to get a balance between the capacity and the quality. The balance means that you can define the parameter based on your request. If you want to get a recovery image, which is the same to

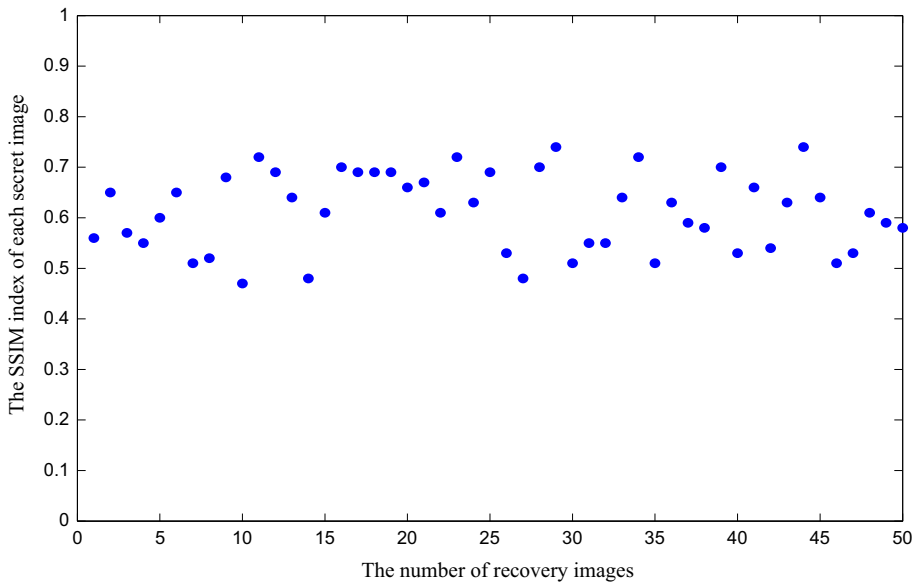

Fig. 7 The SSIM index of 50 recovery images

Table 3 The quality of each secret image under different size of image patches

$P_w \times P_h$	40×30	20×15	8×6	4×3
Secret image 1	0.36	0.36	0.47	0.60
Secret image 2	0.66	0.68	0.74	0.83
Secret image 3	0.62	0.63	0.71	0.81

Table 4 The SSIM index under different sizes of image database

	2M	4M	8M
Secret image 1	0.42	0.43	0.47
Secret image 2	0.63	0.65	0.74
Secret image 3	0.63	0.64	0.71

the secret image, you can divide the secret image into more image patches. However, if you want to hide more information, we can divide the secret image into less image patches (Table 3).

The size of image database also has influence on the quality of the recovery image. If there is a larger image database, the quality of recovery image will be better. Why it happens is that we can get a more appropriate partial duplicate to represent the secret image patch. It means that the duplicate is more similar to the original image patch. In this experiment, we use the number of image patches in each category to represent the different size of image database. And the number of image patches we set is 2 million, 4 million, and 8 million. Table 4 shows the SSIM index under different sizes of image database.

4.3 The security of our approach

This section will analyze the security of our steganography method in two aspects: resistance to steganalysis tools and the security to attackers.

The resistance to the steganalysis tools As we all know, there are many kinds of steganalysis tools such as (Ker 2004, 2005), which are working on the modification traces. It is generally known that an ideal image steganography method has a perfect resistance to all kinds of steganalysis tools. It is unfortunate that the existing image steganography methods embed the secret information into the cover image by modifying the content or the structure of the cover. Consequently, it is possible for steganalysis tools to detect the steganography through the modification traces left in the cover image. However, our approach is not sensitive to the existing steganalysis tools. The reason for this is that our approach find the original image, which contains the similar image patch as the stego-image instead of modifying the content or structure of the cover image. In other words, our approach embeds the secret image into the cover image without any modification traces. Therefore, if we use our approach to implement steganography, the secret image in the cover image will not be detected by the existing steganalysis tools.

The security to attackers In this section, we will discuss the probability of getting the secret information from those stego-images. We assume that attackers can capture those communicated stego-images. For an image as the secret information, if we get one-third of the correct image patches, the complete image can be guessed easily.

Fig. 8 The attack ways on the recovery image

Table 5 The sample rate error rate (SCR) of different attacks

Attack	Luminance change(%)	Contrast change (%)	JPEG compression (%)	Gaussian noise adding (%)
Secret image1	0.1	0.9	1.5	44.6
Secret image 2	0.4	3.0	0.8	68.5
Secret image 3	0.6	1.0	0.5	64.9
Average SCR	0.37	1.63	0.93	59.33

First, an attacker captures the stego-images to extract the secret information. If attackers want to get the secret information, the secret key and the location information are necessary. However, the secret key will not be known except the sender and receiver side.

The computation of taking brute force to crack the secret information may be heavier, even if attackers know the secret key and location information. When we divide the secret image into $\|PB\|$ patches, the total number of the different image stitching ways offered is Pb which can be calculated using (7). However, this case only happens in that attacker knows the secret key and the location information. If attackers do not know the secret key and location information, there are $\|PB\|$ image patches which can be considered as the stego-image patches. Hence, Pb can be calculated using (8).

$$Pb = \frac{\|PB\|!}{\left(\|PB\| - \left\lceil \|PB\|/3 \right\rceil\right)!} \tag{7}$$

$$Pb = \frac{\|PB\|!}{\left(\|PB\| - \left\lceil \|PB\|/3 \right\rceil\right)!} \times \|PB\|^{\lceil \|PB\|/3 \rceil} \tag{8}$$

We present an example to provide more insight for the security. Suppose $\|PB\| = 9$, and this means that the image will be divided into 9 patches. We assume that an attacker knows the secret key and the location information. There are 504 kinds of image stitching ways. If they do not know the secret key, there are 367,416 kinds of image stitching ways. Therefore, it is hard to find a correct one from those ways. Therefore, we have confirmed that our approach is secure.

4.4 The robustness to image attacks

In this section, we will test the robustness of our approach to the image attacks, such as luminance change, contrast enhancement, JPEG compression, and Gaussian noise. We will use these attack ways on the recovery images. Figure 8 shows the attack on the recovery images, which are recovered by our approach.

1. Luminance change by adding the intensity of image pixels with 15;
2. Contrast change by multiplying the intensity of the image pixels with a fact of 1.4;
3. JPEG compression with a fact of 90%;
4. Gaussian noise adding with default values.

We will use the SSIM index change rate (SCR) to verify the robustness of different ways of each image attack. We assume that the SSIM between the secret image and recovery image before we attack the recovery image is S_{before}, and the SSIM index between the secret image and recovery image after we attack the recovery image is S_{after}. Therefore, the SCR can be calculated by using (9). Table 5 shows the results. From the table, we can find that our approach has the lower SCR to all of attacks we employed except Gaussian noise. That means our approach is more robust.

$$\frac{S_{\text{before}} - S_{\text{after}}}{1} \times 100\% \tag{9}$$

5 Conclusion and future work

This paper proposes a novel method to hide the secret image in the natural image. Given an original image divided into a number of image patches, our approach can find the approximately similar image patches to present the secret image patches by comparing the features of those patches. As a result, each partial-duplicate image can hide one or sev-

eral secret image patches. Thus, the hiding capacity of the proposed method is much higher than those of the existing coverless image steganography methods. Since we use the partial-duplicate images that are natural images as stego-images, our approach can resist all of existing steganalysis tools. Meanwhile, our approach achieves good security, as it uses a key to control and decide which part of a natural image is used for hiding secret information. Our possible future work will extend our image database to improve the quality of the recovered secret image. We also pay more attention on the increase of hiding capacities.

Acknowledgements This work was supported, in part, by the National Natural Science Foundation of China under grant numbers 61602253, U1536206, 61232016, U1405254, 61373132, 61373133, 61502242, 61572258, and 61672294; in part, by the Jiangsu Basic Research Programs-Natural Science Foundation under grant numbers BK20150925 and BK20151530; in part, by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund; and in part, by the Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET) fund, China.

Compliance with ethical standards

Conflicts of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Cao Y, Zhou Z, Sun X, Gao C (2018) Coverless information hiding based on the molecular structure images of material. Comput Mater Continua 54(2):197–207

Chang C, Kieu T, Chou Y (2009) Reversible information hiding for VQ indices based on locally adaptive coding. J Vis Commun Image Represent 20:57–64

Cheddad A, Condell J, Curran K, Kevitt M (2010) Review: 'Digital image steganography: survey and analysis of current methods'. J Signal Process 90:727–752

Chen W (2007) Color image steganography scheme using set partitioning in hierarchical trees coding, digital Fourier transform and adaptive phase modulation. Appl Math Comput 185:432–448

Chen B, Zhou C, Jeon B, Zheng Y, Wang J (2017a) Quaternion discrete fractional random transform for color image adaptive watermarking. Multimed Tools Appl. https://doi.org/10.1007/s11042-017-5511-2

Chen X, Chen S, Wu Y (2017b) Coverless information hiding method based on the Chinese character encoding. J Internet Technol 18(2):313–320

Efros A, Freeman W (2001) Image quilting for texture synthesis and transfer. In: Conference on computer graphics and interactive techniques, New York, USA, vol 2001, pp 341–346

Fridrich J, Kodovsky J (2012) Rich models for steganalysis of digital images. IEEE Trans Inf Forensics Secur 7:868–882

Hioki H (2002) A data embedding method using BPCS principle with new complexity measures. In: Proceedings of Pacific Rim Workshop on digital steganography, pp. 30–47

Johnson N, Jajodia S (1998) Exploring steganography: seeing the unseen. Computer 31:26–34

Kawaguchi E (2005) BPCS-steganography principle and applications. In: Knowledge-based intelligent information and engineering systems, Melbourne, Australia. vol 3684, pp 289–299

Ker A (2004) Improved detection of LSB steganography in grayscale images. Springer, Heidelberg, pp 97–115

Ker A (2005) Steganalysis of LSB matching in grayscale images. IEEE Signal Process Lett 12:441–444

Kieu M, Lai K, Tran T, Le T(2016) A fusion of bag of word model and hierarchical K-Means++ in image retrieval. In: International symposium on integrated uncertainty in knowledge modelling and decision making, Springer, Cham, pp 397–408

Liao X, Qin Z, Ding L (2017a) Data embedding in digital images using critical functions. Sig Process Image Commun 58:146–156

Liao X, Yin J, Guo S, Li X, Sangaiah A (2017b) Medical JPEG image steganography based on preserving inter-block dependencies. Comput Electr Eng. https://doi.org/10.1016/j.compeleceng.2017.08.020

Li Z, Chen X, Pan X, ZX (2009) Lossless data hiding scheme based on adjacent pixel difference. In: International conference on computer engineering and technology. IEEE Computer Society, Washington Vol 1, pp 588–592

Li X, Wang J (2007) A steganographic method based upon JPEG and particle swarm optimization algorithm. Inf Sci Int J 177:3099–3109

Luo W, Huang F, Huang J (2010) Edge adaptive image steganography based on LSB matching revisited. IEEE Trans Inf Forensics Secur 5:201–214

Mckeon R (2007) Strange Fourier steganography in movies. In: IEEE international conference on electro/information technology, Chicago, USA. vol 2007, pp 178–182

Mielikainen J (2006) LSB matching revisited. IEEE Signal Process Lett 13:285–287

Sivic J, Zisserman A (2003) Video Google: A text retrieval approach to object matching in videos. Proceedings of the 9th IEEE International Conference on Computer Vision, pp. 1470-1477

Sivic J, Zisserman A (2012) Video Google: a text retrieval approach to object matching in videos. In: IEEE international conference on computer vision, vol 7, pp 1470–1477

Sun H, Grishman R, Wang Y (2017) Active learning based named entity recognition and its application in natural language coverless information hiding. J Internet Technol 18(2):443–451

Wang Z, Bovik A, Sheikh H, Simoncelli E (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13:600–612

Wang J, Lian S, Shi Y (2017) Hybrid multiplicative multi-watermarking in DWT domain. Multidimension Syst Signal Process 28(2):617–636

Wu K, Wang C (2014) Steganography using reversible texture synthesis. IEEE Trans Image Process 24:130–139

Wu K, Wang C (2015) Steganography using reversible texture synthesis. IEEE Trans Image Process 24:130–139

Wu H, Wu N, Tsai C, Hwang M (2005) Image steganographic scheme based on pixel-value differencing and LSB replacement methods. Proc Vis Image Signal Process 152:611–615

Xia Z, Li X (2017) Coverless information hiding method based on LSB of the character's unicode. J Internet Technol 18(6):1353–1360

Xiong L, Xu Z, Shi Y (2017) An integer wavelet transform based scheme for reversible data hiding in encrypted images. Multidimension Syst Signal Process 29(3):1191–1202

Yin Z, Niu X, Zhang X, Tang J, Luo B (2017b) Reversible data hiding in encrypted AMBTC compressed images. In: Multimedia tools and applications, pp 436–445

Yin Z, Niu X, Zhou Z, Tang J, Luo B (2017a) Improved reversible image authentication scheme. Cogn Comput 8(5):890–899

Yuan C, Xia Z, Sun X (2016) Coverless image steganography based on SIFT and BOF. J Internet Technol 18:209–216

Zhou Z, Sun X, Chen X, Chang C, Fu Z (2014) A novel signature based on combination of global and local signatures for image copy detection. Secur Commun Netw 7(11):1702–1711

Zhou Z, Sun H, Harit R, Chen X, Sun X (2015) Coverless image steganography without embedding. In: International conference on cloud computing and security, Nanjing, China, pp 123–132

Zhou Z, Cao Y, Sun X (2016a) Coverless information hiding based on bag-of-words model of image. J Appl Sci 34:527–536

Zhou Z, Wang Y, Wu Q, Yang C, Sun X (2016b) Effective and efficient global context verification for image copy detection. IEEE Trans Inf Forensics Secur 12:48–63

Zhou Z, Yang C, Chen B, Sun X, Liu Q, Wu Q (2016c) Effective and efficient image copy detection with resistance to arbitrary rotation. IEICE Trans Inf Syst E99.D(6):1531–1540

Zhou Z, Wu Q, Yang C, Sun X, Pan Z (2017a) Coverless image steganography based on histograms of oriented gradients-based hashing algorithm. J Internet Technol 18(5):1177–1184

Zhou Z, Wu Q, Huang F, Sun X (2017b) Fast and accurate near-duplicate image elimination for visual sensor networks. Int J Distrib Sens Netw 13(2):1–12