J. Vis. Commun. Image R. 30 (2015) 75–85

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

An efficient forgery detection algorithm for object removal by exemplar-based image inpainting ☆

Zaoshan Liang [a], Gaobo Yang [a,*], Xiangling Ding [a], Leida Li [b]

[a] School of Information Science and Engineering, Hunan University, Changsha 410082, China
[b] School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China

ARTICLE INFO

Article history:
Received 21 October 2014
Accepted 19 March 2015
Available online 28 March 2015

Keywords:
Image forensics
Blind detection
Object removal
Exemplar-based inpainting
Central pixel mapping
Greatest zero-connectivity component labeling
Fragment splicing detection
Load factor

ABSTRACT

As a popular image manipulation technique, object removal can be achieved by image-inpainting without any noticeable traces, which poses huge challenges to passive image forensics. The existing detection approach utilizes full search for block matching, resulting in high computational complexity. This paper presents an efficient forgery detection algorithm for object removal by exemplar-based inpainting, which integrates central pixel mapping (CPM), greatest zero-connectivity component labeling (GZCL) and fragment splicing detection (FSD). CPM speeds up suspicious block search by efficiently matching those blocks with similar hash values and then finding the suspicious pairs. To improve the detection precision, GZCL is used to mark the tampered pixels in suspected block pairs. FSD is adopted to distinguish and locate tampered regions from its best-match regions. Experimental results show that the proposed algorithm can reduce up to 90% of the processing time and maintain a detection precision above 85% under different kinds of object-removed images.

1. Introduction

The prevalence of powerful image processing softwares and the advancement in digital cameras have given rise to large amounts of doctored images with no obvious traces, triggering a great demand for automatic forgery detection algorithms that can identify the trustworthiness of a candidate image [1]. Especially, passive image forensics has attracted great research interests since it does not require any auxiliary data such as watermarks or signatures [2,3]. Up to present, there are extensive works for the passive forensics of various tampering such as sub-sampling [4], double jpeg compression [5] and median filtering [6].

For digital images, the manipulations of image objects including object adding, removing or modifying are of the most attention because these changes of objects will directly mislead the understanding and awareness of the image content. In general, image object removal techniques can be grouped into two categories [2]: copy-move and image inpainting. Copy-move is achieved by copying a region from an image and then pasting it into the same image with the intent of hiding undesired objects [3]. Due to its simplicity, copy-move has become the most widely used method for manipulating the semantics of an image. Recently, extensive researches have been done on the passive forensics against copy-move and a series of detection algorithms have been presented [7–13].

In the past few years, image inpainting has made great progress and is now playing an important role in contents correction and image restoration. However, it can also be a useful tool for object removal [2]. Inspired by real techniques for painting restoration, image inpainting methods fill the holes left by object removal through exploiting the information preserved in the surrounding regions. Object removal achieved by image inpainting can preserve texture and structure continuity [14]. As a result, it leaves no obvious traces of tampering, which makes passive image forensics extremely challenging. Up to now, little study has been done on passive forensics for image inpainting, among which there are three representative works. As a first attempt, Wu et al. [15] proposed a passive forensics method to discriminate natural images from inpainted images, which used zero-connectivity labeling to yield matching degree of the blocks in suspicious regions and computed the fuzzy memberships to identify the tampered regions by a cut set. However, it is a semi-automated detection method which requires to manually select a region of suspicion in advance. Moreover, the algorithm seeks for suspicious blocks by means of full search, resulting in high computation complexity. Later, Bacchuwar and Ramakrishnan [16] devised an improved method

☆ This paper has been recommended for acceptance by M.T. Sun.

* Corresponding author. Fax: +86 0731 8882 1907.
E-mail address: yanggaobo@hnu.edu.cn (G. Yang).

http://dx.doi.org/10.1016/j.jvcir.2015.03.004
1047-3203/

based on the luminance component of the image and median comparison of the blocks in the region of suspicion. The improved approach converted the image to YUV space and only the Y component was used to search for suspicious blocks. Before calculating matching degree of block pairs, the medians of blocks were compared. If the difference was great enough, the calculation of matching degree would be skipped. Though the simplified method can reduce processing time to some extent, it is still a semi-automatic approach. In order to overcome the shortcoming of abovementioned methods, Chang et al. [17] presented an automatic forgery detection algorithm for exemplar-based inpainting images using multi-region relation. In this method, zero-connectivity feature was also used to search for suspicious blocks and vector filtering was exploited to remove the false-alarmed blocks located in the uniform background. Besides, three types of regions were produced according to the relationships among suspicious regions: multi-link, single-link and self-link, while only the multi-link regions were regarded as tampered. Furthermore, weight-transformation based mapping approach was adopted to accelerate the search of suspicious blocks. In spite of outperforming the former two methods in terms of computational efficiency, weight-transformation based mapping approach cannot optimize load factor and search range simultaneously, which thereby constrains further improvement of search speed and detection accuracy. Therefore, we propose an improved passive forensics method, which enables an optimization between the load factor and the search range.

This paper proposes an efficient passive forgery detection method for object removal by exemplar-based inpainting using central pixel mapping (CPM), greatest zero-connectivity component labeling (GZCL) and fragment splicing detection (FSD). As compares to the current state of the art [15–17], the contributions of this paper are as follows: Firstly, CPM is proposed to speed up suspicious block search, which assigns hash value for target block according to the color information of central pixel and then searches for the best-match block among those with similar hash values. The hash values obtained from CPM can better represent the color distributions of image blocks, producing a higher aggregation of similar blocks, which further narrowing the search range. Meanwhile, CPM generates one to one mapping between image blocks and hash values, so that the load factor of the hash table is more close to 1, thus the search efficiency is further improved. Above all, CPM enhances search performance in terms of both load factor and search range. Secondly, GZCL is adopted to mark the tampered pixels in suspicious blocks. After searching the suspicious blocks, the existing methods directly label all the pixels in suspicious block as being tampered, which will lead to higher false alarm and a jagged edge of the detected region. In order to improve detection precision, GZCL takes advantage of the greatest zero-connectivity component in the difference array and marks the corresponding pixels within the suspicious block as being tampered, through which not only a smoother edge is obtained but also the false positive rate will be reduced. Thirdly, FSD is used to distinguish tampered regions from reference regions and obtain the final position of forgery. Most existing algorithms consider both reference and tampered regions as being forged, which makes the reference regions to be another reason of false alarm. In order to solve this problem, FSD calculates the number of regions matched with the target region and filters out those with less matched regions, resulting in a higher detection precision. Experimental results show that the proposed algorithm can reduce 90% of processing time and keep false alarm lower than 15%.

The rest of this paper is organized as follows: Section 2 briefly introduces the image inpainting techniques, particularly exemplar-based inpainting. Section 3 describes the proposed detection method. Section 4 focuses on the fast search algorithm based on central pixel mapping. Section 5 presents the experimental results and analysis. Finally, we conclude in Section 6.

2. Image inpainting

Image inpainting is a technique used for image restoration that can recover the lost information in old photographs and remove scratches in images. However, it could also be exploited to remove image semantic objects for malicious motives. In this case, image inpainting becomes a forgery manipulation. Before we go into details about forensics against inpainting based object removal, it is necessary to briefly introduce the principles of image inpainting and analyze the possible traces left behind.

Based on their application in image restoration, inpainting techniques can be mainly divided into two categories [14]. One is pixel-based approaches [18–20], which are mainly focused on structural repairing and used to repair small-scale defects (such as cracks, and scratches). However, it will generate obvious blur when the damaged region is large or the texture is rich. The other one is exemplar-based approaches [21–23], which combine structure recovery with texture repairing and can be used to restore larger loss of information in the image. Criminisi's algorithm [21] is among the most popular exemplar-based inpainting approaches. Many researchers have exploited the inpainting framework in Criminisi's algorithm due to its good visual effect. Therefore, we take Criminisi's algorithm as an example and introduce the principle of exemplar-based inpainting.

Before inpainting, users need to select target region to be removed and filled, as shown in Fig. 1(a). The target region is indicated by Ω, and its contour is denoted by $\partial\Omega$ and the source region, Φ, may be defined as the entire image minus the target region. Then the inpainting procedure is conducted as follows.

1. Compute the priorities of the points along $\partial\Omega$ and find the point p with highest priority. Then, the block Ψ_p centered at the point p is selected as target block, as shown in Fig. 1(b).
2. Search for the reference block Ψ_q which is most similar to Ψ_p in source region Φ, as shown in Fig. 1(c).
3. Fill the area to be inpainted in Ψ_p using the corresponding pixels of Ψ_q, and update the priorities among Ψ_p as well as $\partial\Omega$, as shown in Fig. 1(d).
4. Repeat step 1 to step 3 until the target region is entirely filled.

By integrating texture synthesis with pixel-based approaches, the algorithm performs well in terms of both perceptual quality and efficiency. Subsequently, there are some improvements on Criminisis algorithm, which mostly concentrate on improving priority calculation and optimizing patch searching method [22,23]. Liu and Caselles [22] proposed a novel inpainting method based on multi-scale Graph Cuts, which ensured the continuity of reconstruction at the boundary of the inpainted region as well as visually coherent reconstruction inside the hole. Wang et al. [23] introduced a regularized factor into the confidence term to lessen descending effect, which significantly improved the filling order. Meanwhile, for the condition when multiple candidates were found during the best-match reference block searching, a two-round search strategy based on a modified sum of squared differences (SSD) and normalized cross correlation (NCC) was proposed to make the inpainting more robust to images with large removal regions.

Although there are differences in ways of calculating priority and searching for best-match block between Criminisi's algorithm and improved algorithms, filling the target block with its best-match block will introduce abnormal similarity. However, there exists large amount of irregular pieces in natural images, which

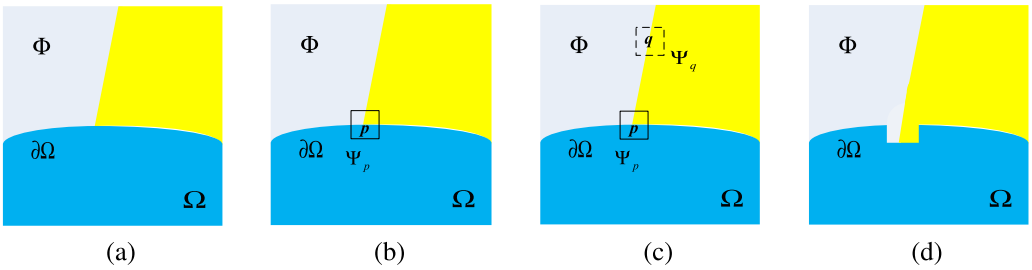

Fig. 1. Exemplar-based inpainting. (a) Specify target region Ω, (b) select target block Ψ_p, (c) search for reference block Ψ_q and (d) update priority and contour.

share seemingly the same textured surface, but in fact are different [15]. In the imaging process, these differences will result in unequal pixel values. Therefore, the filling scheme of exemplar-based inpainting cannot represent subtle differences in natural scene, but lead to abnormal similarity between blocks, which provides clues for image forensics.

3. Forgery detection algorithm for exemplar-based image inpainting

The purpose of the proposed method is to detect object removal by exemplar-based image inpainting. According to the abnormal similarity of block pairs described above, a passive forensics algorithm based on CPM, GZCL and FSD is proposed. Fig. 2 shows the framework of the proposed algorithm, which consists of four procedures: Firstly, zero-connectivity feature is applied to suspicious block searching which is usually the most time-consuming part. In order to improve search efficiency and maintain good detection results, this paper presents a novel fast search algorithm based on CPM. Secondly, GZCL is used to mark the tampered pixels in suspicious blocks. Thirdly, vector filtering is adopted to remove the false-detected regions in uniform background. Finally, FSD is employed to filter out the reference regions and get the final locations of tampered regions.

3.1. Suspicious block searching based on zero-connectivity feature

In exemplar-based inpainting, the unknown part of the target block is filled using the corresponding pixels of reference block, resulting in a number of zeros connecting in the difference array between target block and its reference block, that is, a zero is 8-neighborhood connected with other zeros. This characteristic is named as zero-connectivity feature [15]. Due to its simplicity and effectiveness, zero-connectivity feature is also adopted in this paper to search for suspicious blocks.

For each block Ψ_p in the test image I where p is its upper left point, we search for its reference block $\Psi_{\widehat{q}}$ which is most similar to Ψ_p across the complement of Ψ_p in the image $\Phi = \overline{\Psi_p}$, which can be formulated as

$$\Psi_{\widehat{q}} = \arg\max_{\Psi_q \in \Phi} n(\Psi_p, \Psi_q) \tag{1}$$

where $n(\Psi_p, \Psi_q)$ is the matching degree of the block pair.

The matching degree is computed following four steps. Firstly, one block is subtracted from another to get the absolute values of differences for R, G and B components, which are denoted by Δ_R, Δ_G and Δ_B respectively. Then, Δ_R, Δ_G and Δ_B are converted to binary difference arrays D_R, D_G and D_B respectively. Next, mathematical "OR" is performed on D_R, D_G and D_B producing the difference array D_{pq}. At last, the greatest zero-connectivity component which contains the largest number of "0" is taken as the matching degree of the block pair.

In order to find the zero-connectivity components in D_{pq}, an 8-connectedness labeling method is used to extract zero-connectivity features. Fig. 3 illustrates how to compute the matching degree, where R_p, G_p, B_p, R_q, G_q and B_q represent the R, G, B components of target block and reference block respectively, while the zero-connectivity components are labeled with green[1] lines. Here, the matching degree is 8.

When the matching degree between target block Ψ_p and reference block $\Psi_{\widehat{q}}$ meets Eq. (2), the pair of blocks will be determined as suspicious, and the matching relationship between them will be record as a similarity vector, which starts from Ψ_p and points to $\Psi_{\widehat{q}}$.

$$\frac{n(\Psi_p, \Psi_{\widehat{q}})}{E^2} \geqslant \eta \tag{2}$$

where E is the block length, η is a threshold decided by fussy membership (see details in [15]).

3.2. Greatest zero-connectivity component labeling

After suspicious block searching, the pixels within those blocks need to be labeled to generate a group of suspicious regions. The existing methods [15–17] adopt a whole block labeling (WBL) strategy, where all pixels in suspicious blocks are marked as being tampered. However, we observe that filling operation is only for unknown portion of the target block during the inpainting process. This means that a suspicious block contains both tampered pixels and original pixels. This phenomenon is most pronounced on the edge of the object removal region. Hence, WBL would mistake original pixels as being tampered, thereby cause high false positive rate and get jagged edges, which will definitely decrease the precision of forgery location.

In order to remove the false alarm caused by WBL and get a more accurate location of suspicious region, this paper proposes a new method based on GZCL. First, we find the greatest zero-connectivity component in the difference array $D_{\widehat{pq}}$ obtained by suspicious block searching. Then, for suspicious block Ψ_p, the corresponding pixels are marked as being tampered which share the same position with the greatest zero-connectivity component, as shown in Eq. (3):

$$label(\Psi_p(x,y)) = \begin{cases} tampered, & D_{\widehat{pq}}(x,y) \in GZC \\ authetic, & otherwise \end{cases} \tag{3}$$

where (x, y) is a point in the block, GZC refers to the greatest zero-connectivity component.

Fig. 4 compares the results of GZCL and WBL, where Fig. 4(a) is the mask of inpainting, Fig. 4(b) is the result of WBL and Fig. 4(c) is the result of GZCL. It can be seen from Fig. 4(b) that WBL gets a

[1] For interpretation of color in Figs. 3–5, 7, 8, 13 and 14, the reader is referred to the web version of this article.

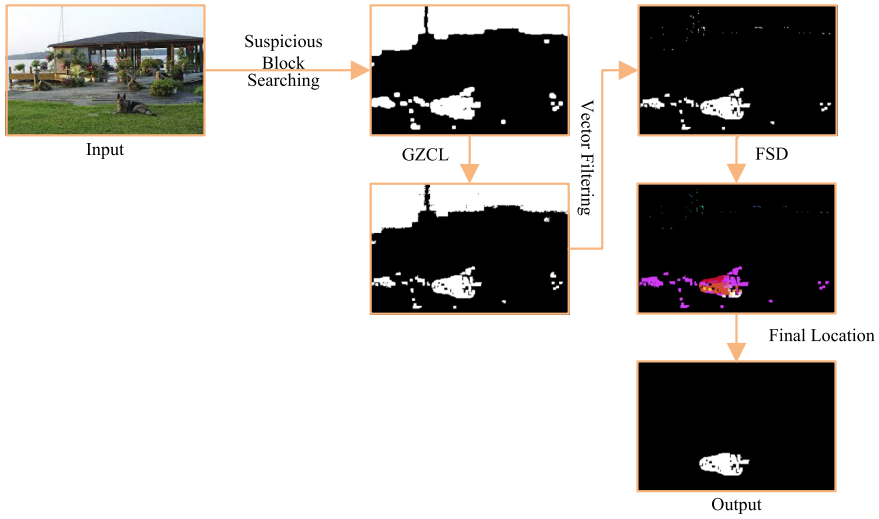

Fig. 2. Framework of the proposed algorithm.

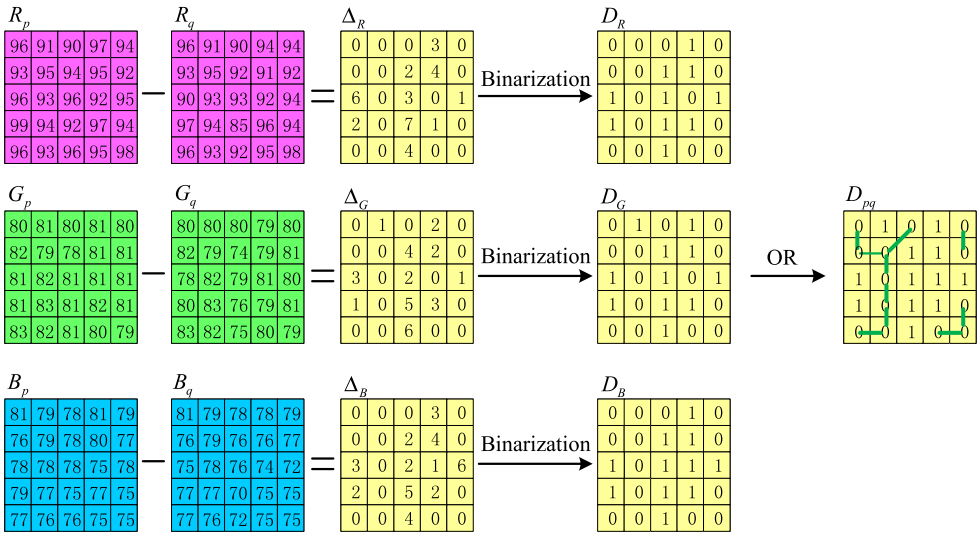

Fig. 3. Matching degree computation.

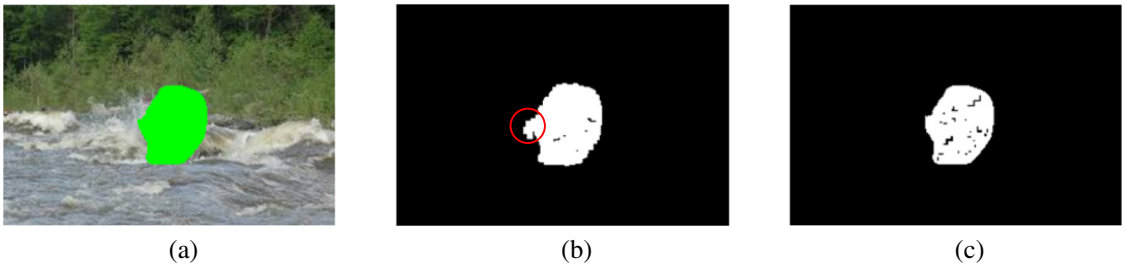

Fig. 4. Comparison between WBL and GZCL. (a) Inpainting mask, (b) result of WBL and (c) result of GZCL.

jagged edge. Furthermore, the suspicious region is connected with several mistakenly detected blocks (marked with a red circle). It is clear that a more accurate and smoother edge is obtained by GZCL, as shown in Fig. 4(c). Despite introducing some undetected true positives, GZCL gains a better balance between recall and precision (details are discussed in Section 5).

3.3. Vector filtering

Natural images may have some very similar blocks in the background, especially uniform regions such as the sky or lake, which share high matching degree with its neighboring blocks and will be easily marked as tampered in suspicious block searching. In this

paper, vector filtering [17] is employed to reduce false alarm caused by uniform background, which is based on an analysis for the distribution of similarity vectors. Fig. 5 compares the results with and without vector filtering, where Fig. 5(a) is the original image with blue sky in the background. Large numbers of blocks are detected in suspicious block searching, as shown in Fig. 5(b). Obviously, uniform regions will interfere with the detection result. Therefore, it is necessary to exclude authentic uniform regions from suspicious regions.

Based on vector filtering, three kinds of image areas are defined in this paper: uniform area, textural area and complex area. Each one contains different distribution of similarity vectors. Uniform area contains the similarity vectors with short length, while longer similarity vectors lie in textural area and complex area does not have any similarity vectors. Generally, the inpainted regions are located in the textural area, where similarity vectors have relatively longer length than those of a uniform area. Therefore, the length of similarity vectors can be used to distinguish inpainted regions from uniform regions. Fig. 5(c) shows the result of vector filtering, where most of the false positives caused by uniform regions are removed.

3.4. Fragment splicing detection

A rough location of tampered region can be obtained using the three steps mentioned above. Nonetheless, there exists some reference regions that are falsely identified as being tampered due to the fact that matching relationships between tampered regions and its reference ones are founded during suspicious block search. Therefore, reference regions have become another cause of false alarm. In order to distinguish tampered regions from the reference ones, this paper presents a method based on FSD to remove false positives caused by reference regions, producing the final location of forgery.

By analyzing exemplar-based inpainting, it is not difficult to find that tampered regions are replicated by multiple reference regions, in other words, more than one reference regions are matched to a tampered region. More importantly, the reference regions are distributed as scattered "fragments" and usually matched to only one tampered region. Based on this characteristic, we differentiate tampered regions from its reference ones according to the number of regions matched with a target region. The method is described as follows.

First of all, suspicious region $Region_i$ is assigned with a unique code, as shown in Eq. (4). Secondly, the similarity vectors are used to find reference region $Region_j$ which is best matched with $Region_i$, then the code of $Region_i$ is replaced with the code of $Region_j$, as shown in Eq. (5). The replacement of region code will help count the number of regions matched with a target region. Finally, suspicious regions with less reference regions are filtered out, producing the final position of tampered region.

$$Code(Region_i(x,y)) = code_i \quad i \in [1,\ldots,N] \tag{4}$$

$$RepCode(Region_i(x,y)) = Code(Region_j(x,y)) \quad i,j \in [1,\ldots,N] \tag{5}$$

where (x,y) is a pixel in the region, N is the number of suspicious regions, *RepCode* refers to the replaced region code.

Fig. 6 illustrates the process of FSD. Fig. 6(a) shows the suspicious regions, Fig. 6(b) uses different colors to represent the codes of suspicious regions, Fig. 6(c) reveals the suspicious regions whose codes are replaced with those of its reference regions and Fig. 6(d) shows the final location of tampered region. Obviously, compared with Fig. 6(a), detection accuracy of tampered regions improves significantly.

4. Fast search algorithm based on central pixel mapping

In Section 3.1, we have briefly introduced the suspicious block searching based on zero-connectivity features. However, the searching process is very time-consuming. Given an image with size $M \times N$ and a search window with size $E \times E$, the matching degree has to be computed $[(M-E+1)(N-E+1)]^2$ times if full search is adopted. Moreover, the time complexity of matching degree computation is $O(E^2)$. Obviously, the computation complexity of full search is extremely high. Therefore, a fast search algorithm is highly desired. To this end, several methods have been proposed to speed up suspicious block searching [17,24]. In [24], a randomized searching method was used to increase the computational efficiency, but it may fall into the local optimization. Then, Chang et al. [17] presented a weight transformation based searching strategy which reduced the searching time and maintained a relatively better detection accuracy. However, it could not optimize load factor of hash table and search range (detail definitions are in Sections 4.3.1 and 4.3.2) simultaneously, thereby constrained further improvement of search speed and detection accuracy. In order to enhance both load factor and search range, this paper proposes a novel search algorithm based on CPM, which transforms the color information of central pixel in target block into a hash value and search for the best-match block in those blocks with similar hash values.

4.1. Block mapping technique

The basic idea of block mapping is as follows. Firstly, a hash function is constructed to extract color information of the blocks in the image. In order to speed up the search process, matching degree computation between target block and "alien" blocks, which are entirely different from the target one, should be avoided, and gathering blocks with similar color distribution is necessary. As a result, the hash function also needs to follow this principle, assigning similar hash values to similar image blocks. Secondly, the blocks are mapped to a hash table according to their hash values. However, collision will appear when different blocks correspond to the same hash value. To solve this problem, we store the blocks with same hash value in the same list.

Fig. 7 shows the process of block mapping, where each textural region is represented by a unique color, *Value* represents the hash value and the dotted red arrows point to the storage location of the blocks. It is not difficult to find that similar blocks are assigned equal or similar hash values. After mapping, the search of suspicious block consists of two steps: (1) locate the target block in the hash table; (2) find the best-match block within those with equal or similar hash values. Only the matching degrees among similar blocks are calculated through block mapping, resulting in reduced time complexity and increased search efficiency.

4.2. Central pixel mapping

The key to block mapping is the construction of hash function which will directly affect the search efficiency. In [17], four kinds of hash functions are discussed: equal number matrix transformation, even number matrix transformation, odd number matrix transformation, and prime number matrix transformation. Among them, equal number matrix transformation is the best hash function to extract color information from image blocks. However, it tends to cause a collision which enlarges the load factor of the hash table, while the others can deal with collision but fail to gather the similar blocks together, which in turn expands the search range. Therefore, the above-mentioned functions cannot optimize load factor and search range simultaneously, which

Fig. 5. Vector filtering. (a) Original image, (b) suspicious block search and (c) result of vector filtering.

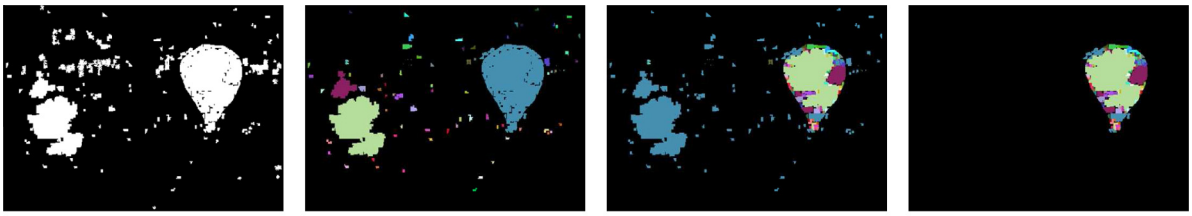

Fig. 6. Fragment splicing detection. (a) Suspicious regions, (b) assign codes for suspicious regions, (c) code replacement and (d) final detection result.

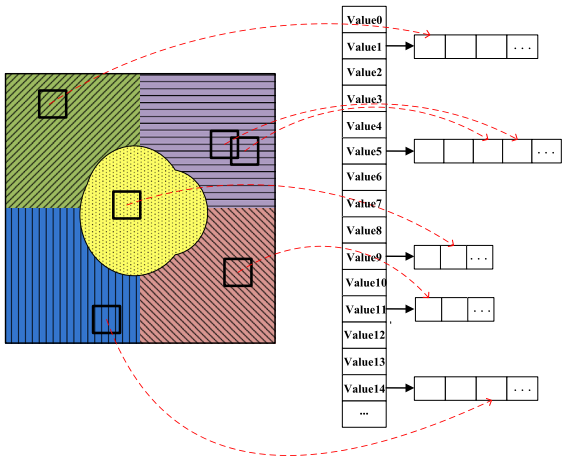

Fig. 7. Block mapping.

constrains further improvement of search speed. In this paper, a new hash function named central pixel mapping (CPM) is defined to gather the blocks with equal value in the central pixel:

$$Key_{\psi_c} = R(c) + G(c) \bullet 2^8 + B(c) \bullet 2^{16} \tag{6}$$

where c is the central pixel of target block, R, G, B represent for the values in red, green and blue channel, respectively.

According to [21], the central pixel of target block is always located at the edge of the region to be inpainted, which means that there exist pixels to be filled in the 8-neighborhood of the central pixel. Similarly, for forgery detection, when the search window has the same size with the inpainting window, there must be tampered pixels in the 8-neighborhood of central pixel. Moreover, the central pixel is also tampered in the case that the search window is smaller than the inpainting window. According to the filling mechanism of exemplar-based inpainting, tampered pixels will share the same values with the corresponding ones in the best-match block. That is to say, the center position of the target block is the same as that of its reference block. Therefore, the search for suspicious blocks can be achieved by only checking the blocks with same central pixels, which verifies that CPM is a proper hash function.

Fig. 8 illustrates the principle of CPM, where "0" represents original pixel, while "1" represents tampered pixel and the inpainting window is 9×9. Fig. 8(a) shows the condition that the search window has the same size with the inpainting window, where a blue box marks the 8-neighborhood of central pixel, a red square represents the central pixel of the block and the tampered pixels within 8-neighborhood are labeled green. It is not difficult to find that there exists tampered pixels in the 8-neighborhood of central pixel. Fig. 8(b) shows the case that the search window is smaller than inpainting window, where the search windows are marked with purple and orange dashed boxes, while the central pixels are represented by red squares. Clearly, the center position of a suspicious block is tampered, which also implies that CPM will not leave any suspicious blocks alone.

4.3. Analysis of CPM

Two features are considered for the construction of hash function: (1) load factor, which denotes the number of blocks represented by a hash value; (2) search range, which denotes the number of hash values compared during the search processing.

4.3.1. Load factor

The measure function of load factor α is defined as follows:

$$\alpha = \frac{N_B}{N_{Key}} \tag{7}$$

where N_B is the number of image blocks and N_{Key} is the number of hash values. According to Eq. (7), the closer to 1 is the load factor, the higher is percent of one-to-one mapping between target blocks and hash values. In this case, less number of matching degree computation is needed, thereby search efficiency is higher.

Table 1 lists the load factors after applying five kinds of block mapping (equal numbers, even numbers, odd numbers, prime numbers and CPM) to a standard test image named "Peppers". It can be seen that prime numbers based method gains a load factor closest to 1, which means that more than half of the blocks are assigned unique hash values. However, it cannot very well represent the color distribution of target block. In addition, CPM produces a sub-optimal load factor in terms of the closeness to 1.

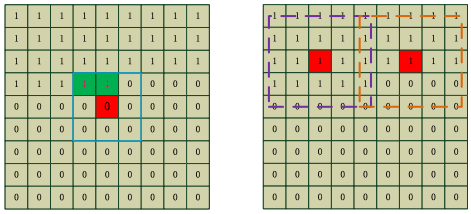

Fig. 8. Principle of CPM. (a) Both search window and inpainting window are 9×9 and (b) search window is 5×5 and inpainting window is 9×9.

Table 1
Load factors of hash function.

Hash functions	Load factors
Prime numbers	1.61
Odd numbers	2.11
Even numbers	3.85
Equal numbers	117.19
CPM	2.05

4.3.2. Search range

Generally, execution time is in proportional to the search range. That is, the smaller the search range is, the less time it consumes. However, we should take both search range and detection accuracy into consideration because that a smaller search range may impact the accuracy of forgery location. Here, we denote detection accuracy as the conformity between the detection result of block mapping and that of full search, as shown in Eq. (8). Then, the optimum search range of each hash function is investigated. Similarly, the test is applied on the image used in Section 4.3.1.

$$conformity = \frac{\|R_F \cap R_{BM}\|}{R_F} \tag{8}$$

where R_F is detection result of full search, R_{BM} is the result of block mapping.

Fig. 9 shows the relation between search range and conformity for five kinds of functions mentioned above. It can be seen that CPM gets the minimum search range for a fixed conformity, which is attributed to its highest degree of aggregation for similar blocks. With respect to the prime number transformation, owing the load factor closet to 1 makes the hash value irrelevant to the spatial color distribution; hence, it needs more time and wider search range to find the best-match block for the target one.

In summary, CPM can achieve a better tradeoff between load factor and search range, which can improve the search efficiency

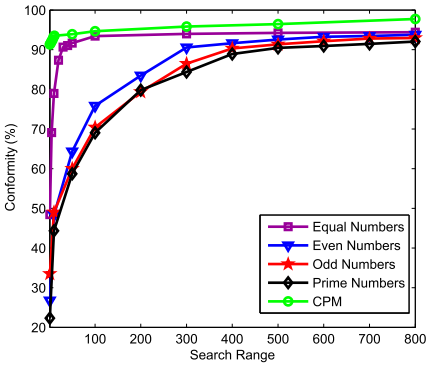

Fig. 9. Search range and conformity.

Table 2
Performance of hash functions.

Hash functions	Search range	Conformity (%)	Execution time (s)
Full search	Whole image	100	7970
Equal numbers	30	90.65	411
Even numbers	300	90.58	382
Odd numbers	400	90.32	378
Prime numbers	500	90.46	403
CPM	1	91.33	67

further. For the same image used in Section 4.3.1, Table 2 compares search range, conformity and execution time of the four kinds of hash functions in [17] and CPM. It can be concluded from Table 2 and Fig. 9 that the minimum execution time is achieved using CPM which reduces more than 99% of the time compared with full search and the conformity of CPM is higher than 90% with a very small search range. Thus, CPM outperforms the other four hash functions.

5. Experimental results

In this section, we first introduce the experimental setup and evaluation metrics. Then, we evaluate the performance of the proposed method by experiments and compare it with three representative methods [15–17].

5.1. Experimental setup

In our experiments, an image database composed of 30 images is created from test images in [17] and general database presented in [25], which is available at http://homepages.lboro.ac.uk/cogs/-datasets/ucid/ucid.html. These images are stored in PNG/TIF format with size ranging from 252×188 to 512×384. In addition, the contents of those images includes landscape, buildings, human, animals, and so on, some of which are shown in Fig. 10. The proposed algorithm is implemented on a laptop computer (Intel(R) Core(TM) i5-2450M CPU @ 2.5 GHz CPU, 4.0 GB RAM) with Visual Studio 2010 and OpenCV library 2.3.1. The experimental parameters are set as follows: block size is 5×5, η is 0.5. The inpainted images are generated using the algorithm in [23] due to the relative better visual quality than existing methods when a large object is removed. To evaluate the robustness and sensitivity of our method, several experiments are conducted in the following scenarios: (1) images with a single object removed without uniform background, (2) images with a single object removed with uniform background and (3) images with multiple objects removed. In addition, the proposed method is also applied to copy-move forged images which are generated by Adobe Photoshop CS4.

5.2. Evaluation metrics

To quantitatively evaluate the performance of the proposed method, recall, precision and average block processing time (ABPT) are defined as follows:

$$\text{Recall} = \frac{N_r}{N_r + N_m} \tag{9}$$

$$\text{Precision} = \frac{N_r}{N_r + N_f} \tag{10}$$

$$\text{ABPT} = \frac{T_I}{N_B} \tag{11}$$

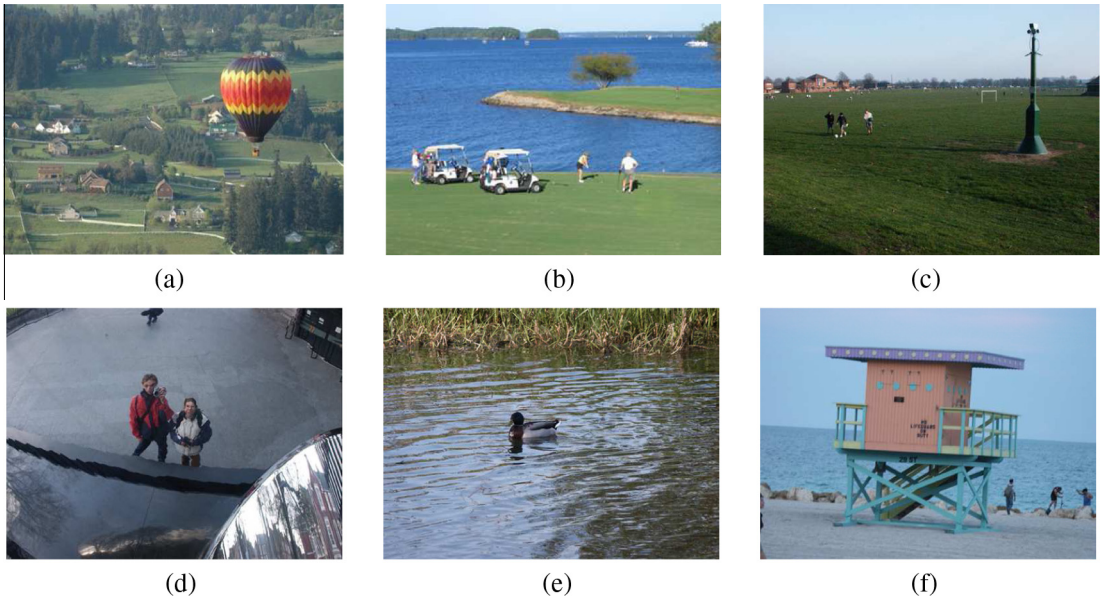

Fig. 10. Test images.

where N_r is the number of pixels correctly detected, N_m is the number of pixels missed, N_f is the number of false positives, $N_r + N_m$ is the number of tampered pixels, $N_r + N_f$ indicates the total number of pixels detected, T_I is the total execution time of image I and N_B indicates the number of blocks in the image.

5.3. Forgery detection against exemplar-based inpainting

5.3.1. Single object removal without uniform background

Fig. 11 shows the detection result for the tampered image with a single object removed, where Fig. 11(a) is the original image taken from [17], Fig. 11(b) is the inpainting mask, Fig. 11(c) is the inpainted image and Fig. 11(d)–(f) are detection results of Wu's method [15], Chang's method [17] and the proposed algorithm, respectively. It can be seen from Fig. 11(d) that the tampered regions contain many false alarms. This is because Wu et al. only uses suspicious block searching and does not take into account the interference of reference regions. Meanwhile, it is not hard to infer from Fig. 11(e) and (f) that Chang's method produces a jagged edge due to the whole blocking labeling, while the edge obtained by our method is much smoother and more accurate with the help of GZCL.

Table 3 compares recall rate, precision, and ABPT of our work, Wu's, Bacchuwar's and Chang's in Fig. 11(c). Compared with Wu's method, the proposed algorithm has reduced execution time

Table 3
Performance under single object removal without uniform background.

Method	Recall (%)	Precision (%)	ABPT (us)
Wu's method [15]	98.00	35.34	30,400
Bacchuwar's method [16]	98.37	40.18	12,400
Chang's method [17]	99.58	91.29	1540
Proposed method	98.52	97.83	252

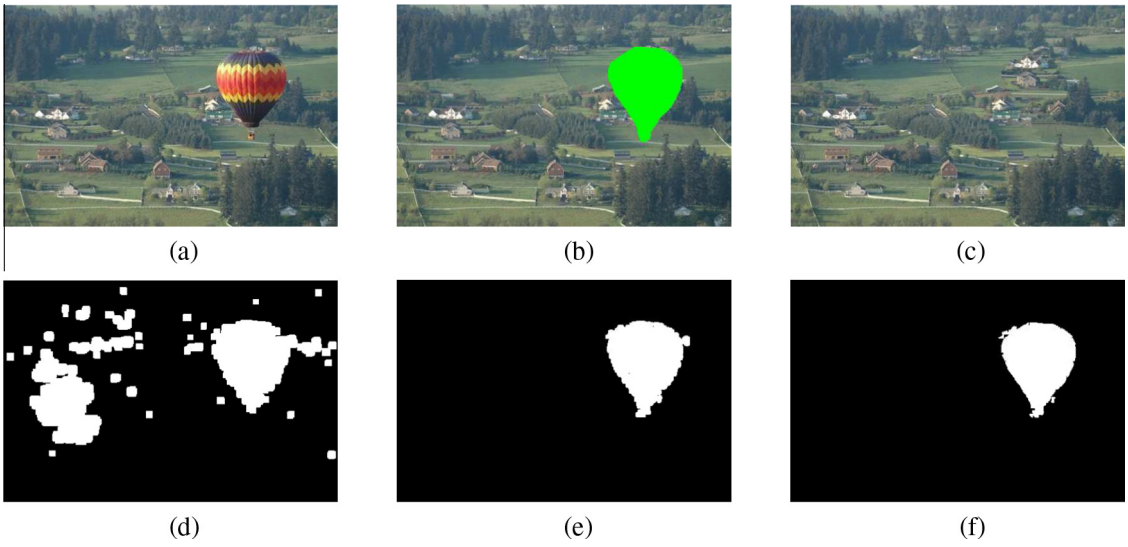

Fig. 11. Single object removal without uniform background. (a) Original image from [17], (b) inpainting mask, (c) inpainted Image, (d) result of Wu's method [15], (e) result of Chang's method [17] and (f) result of the proposed method.

by at least 90% due to CPM. Moreover, the time efficiency of our method is also higher than Bacchuwar's and Chang's method. Even though the recall rate is slightly declined when compared to Chang's method, significant improvements have been made in precision and ABPT, which is a better balance of these three indicators.

5.3.2. Single object removal with uniform background

This experiment shows the performance of forgery detection when the test images contain uniform backgrounds. Fig. 12(a) is the original image taken from [17], Fig. 12(b) is the inpainting mask, Fig. 12(c) is the inpainted image, Fig. 12(d) shows the suspicious regions before vector filtering, Fig. 12(e) shows the suspicious regions with uniform vector filtering, Fig. 12(f) is the result of FSD, while Fig. 12(g)–(h) are the final results of Chang's method [17] and the proposed algorithm, respectively. Fig. 12(d) implies that suspicious block search is heavily affected by the uniform background. In Fig. 12(e), most of false alarms caused by the uniform regions are removed by vector filtering. Fig. 12(f) indicates that the forged region outstands from other regions through FSD. Finally, from Fig. 12(g) and (h), it can be easily seen that Chang's method introduced more false positives than the proposed algorithm.

In this section, only the performances of our work and Chang's are compared because no discussion about object removal under uniform background was made in [15,16]. Similarly, Table 4 analyzes the recall rate, precision, ABPT and time decreasing rate of the two methods. Obviously, the proposed method gain higher time efficiency and precision than Chang's. However, compared with the time indicators in Table 3, the ABPT of two methods has enlarged when the test image contains a uniform background. Due to the similar texture pattern in uniform regions, the load factor of the hash table obtained by block mapping is increased, thereby decreasing the speedup. Nevertheless, there are more influences towards Chang's method where the hash function generates more collisions. Additionally, the uniform background also decreases the precision of forgery region detection.

5.3.3. Forgery detection against multiple objects removal

In this experiment, the performance of the detection against multiple objects removal is verified. Fig. 13(a)–(c) show the original image along with its corresponding actual inpainted regions, inpainting results respectively, while Fig. 13(d)–(e) are the intermediate results of the proposed method and Fig. 13(f)–(g) are final detection results of Chang's method and the presented algorithm. In Fig. 13(f), Chang's method introduces many false alarms around the edge of the object removal region. Although Chang et al. adopt multi-region relation technique to deal with the regions mistakenly detected, but the false areas connected with the tampered regions (marked with red circles) cannot be removed, decreasing the precision rate. But with the use of GZCL, the proposed algorithm can make the false positives less likely melt with the tampered region, thereby improving the detection accuracy, as shown in Fig. 13(g).

Table 4
Performance under single object removal with uniform background.

Method	Recall (%)	Precision (%)	ABPT (us)	Time decreasing rate (%)
Chang's method [17]	99.21	70.64	4348	85.70
Proposed method	96.63	94.20	610	97.10

This experiment is only carried on Chang's and our work because methods of [15,16] only consider the single object removal. Take Fig. 13(c) as a test image, Table 5 demonstrates the performance of our work and Chang's. In spite of slightly lower recall rate than Chang's method, the proposed algorithm has obvious advantages that ABPT is one fifth of Chang's and detection precision is enhanced by 20%.

In order to demonstrate the universality and superiority of the proposed algorithm, two comprehensive experiments are designed: evaluation for the average performance of single object removal and the test on the whole database. In these two experiments, average values of recall rates, precision, ABPT and time decreasing rates are computed and denoted by ARec, APre, AABPT and ATDR, respectively. Table 6 demonstrates that our method achieves the highest detection precision and the least execution time when dealing with single object removal. Meanwhile, it can be concluded from Table 7 that the proposed algorithm cannot only outperform the state-of-the-art method in terms of the overall time efficiency, but also improve the average detection precision above 85% while maintaining a high recall rate.

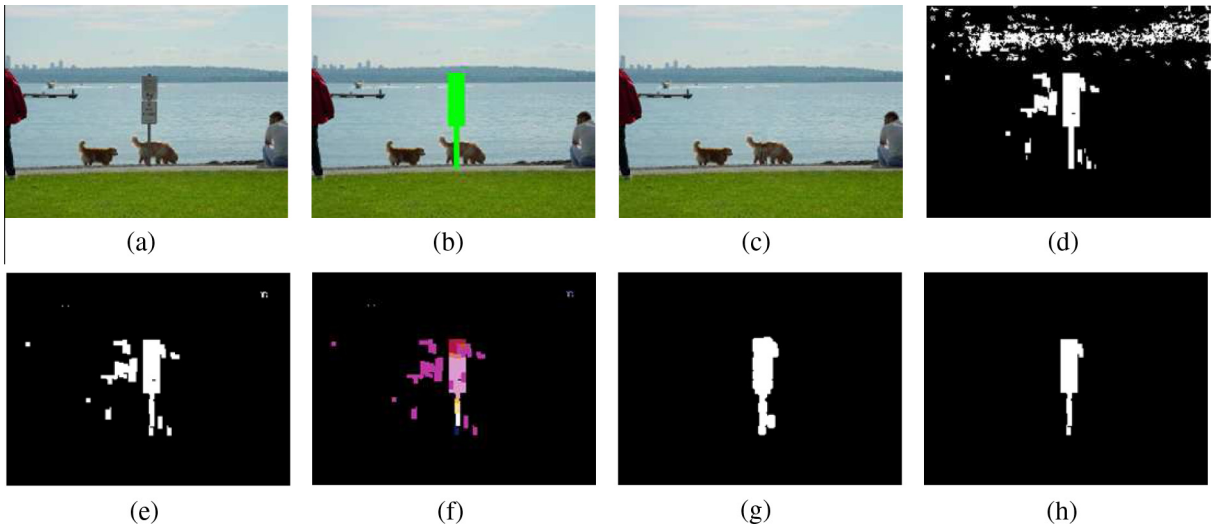

Fig. 12. Single object removal with uniform background. (a) Original image from [17], (b) inpainting mask, (c) inpainted image, (d) result of suspicious block search, (e) result of vector filtering, (f) result of fragment splicing detection, (g) result of Chang's method and (h) result of proposed algorithm.

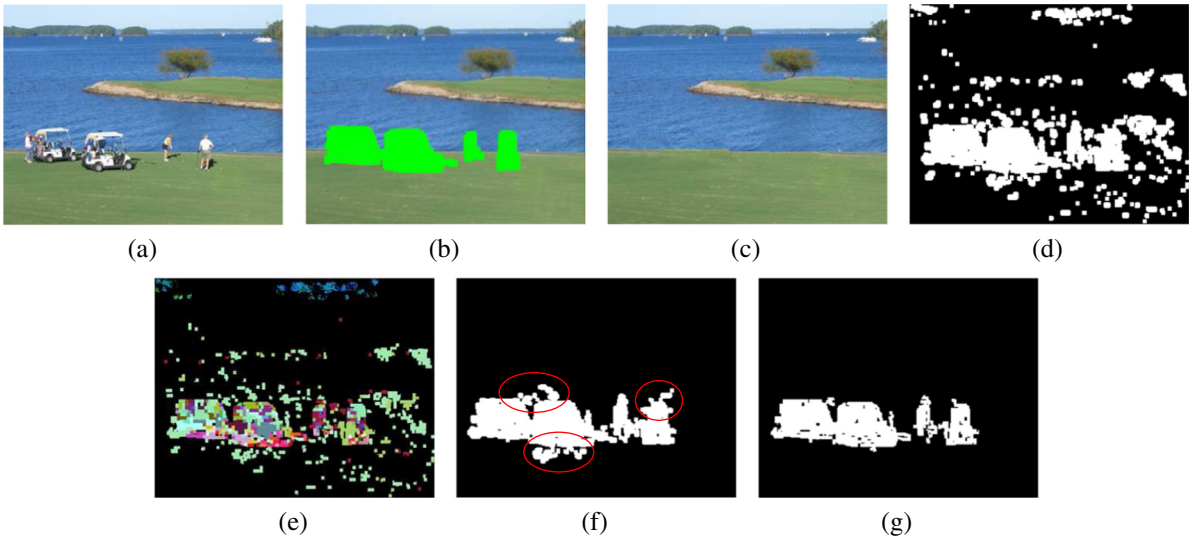

Fig. 13. Forgery detection against multiple objects removal. (a) Original image, (b) inpainting mask, (c) inpainted images, (d) result of suspicious block search, (e) result of fragment splicing detection, (f) result of Chang's method and (g) result of proposed algorithm.

Table 5
Performance under multiple objects removed.

Method	Recall (%)	Precision (%)	ABPT (us)	Time decreasing rate (%)
Chang's method [17]	98.89	71.76	2725	91.04
Proposed method	96.00	91.02	513	97.76

Table 6
The average performance of single objet removal.

Method	ARec (%)	APre (%)	AABPT (us)
Wu's method [15]	96.45	39.08	30,677
Bacchuwar's method [16]	97.02	43.68	12,271
Chang's method [17]	99.63	86.11	1663
Proposed method	97.20	96.72	301

Table 7
The overall performance on the database.

Method	ARec (%)	APre (%)	AABPT (us)	ATDR
Chang's method [17]	98.76	77.33	2984	91.18
Proposed method	96.67	93.13	468	98.46

5.4. Forgery detection against copy-move

Our method is still efficient for copy-move forgery without rotation and zooming, which is a simple and widely used object removal manipulation. Fig. 14 demonstrates the results of forgery detection, where the first column shows the original images taken from [17,25], second column indicates the copy and paste regions by blue and red curves respectively, third column exhibits the tampered images, and the last column presents the results of the proposed approach. Comparing the second column with the last

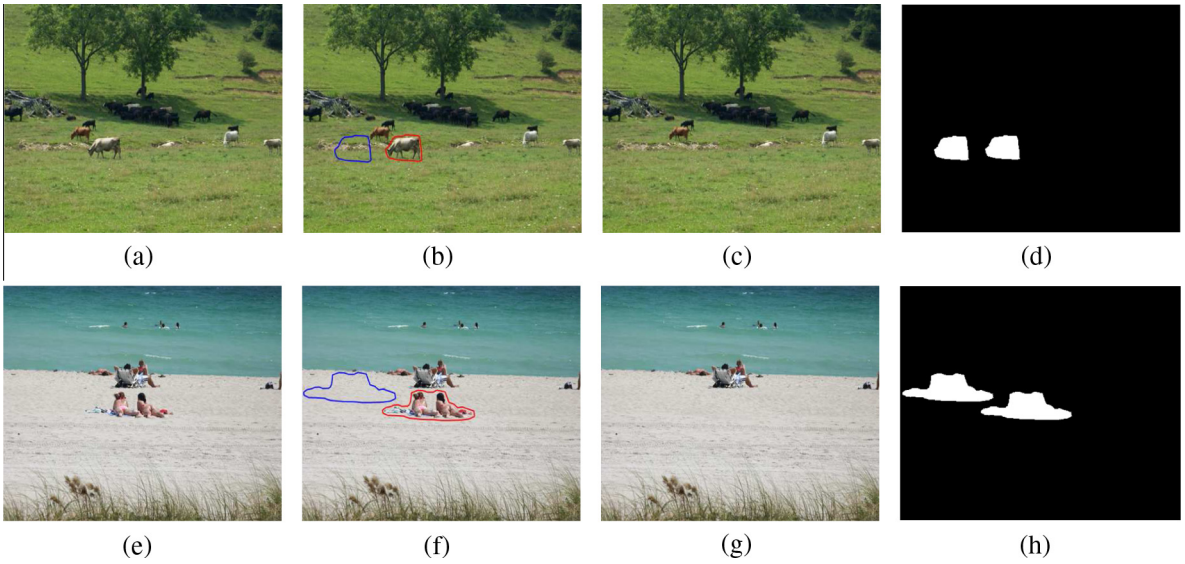

Fig. 14. Forgery detection against copy-move. (a) Original image from [17], (b) copy region and paste region of (a), (c) forged image of (a), (d) detection result of (c), (e) original image from [25], (f) copy region and paste region of (e), (g) forged image of (e), (g) detection result of (h).

column, it is clear that the proposed method can exactly locate the forged areas.

6. Conclusions

A novel passive forgery detection algorithm for object removal by exemplar-based inpainting is proposed based on CPM, GZCL and FSD. CPM reveals the relationship between inpainted blocks and the referenced ones, accelerating the suspicious block search in terms of load factor and search range. GZCL and vector filtering are used to get the location of suspicious regions. Moreover, the adoption of FSD technique accurately identifies the forged regions from suspicious regions. Our experiments show that the proposed algorithm reduces more than 90% of the processing time when compared with the traditional algorithms, while it maintains a precision above 85%. Besides, our method also performs well when detecting copy-move forged images.

However, there are still some limitations in this work. First, an assumption is made that the tampered images are generated by the exemplar-based image inpainting. Thus, the proposed approach is incapable of detecting object removal achieved by other techniques like image seam carving. Second, since the exemplar-based inpainting is in nature to fill the gap by searching similar patches within an image, the proposed approach depends on the similarities of patches for tampering detection. If the forged region is further processed by some post-processing techniques like re-compression, low-pass filtering and blurring, the proposed approach may fail. Because post-processing can also be regarded as a forgery operation, this is actually a detection of multiple forgery operations. The ambiguous processing artifacts left by them should be separated if possible. In the future, we will further investigate this issue.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (61072122, 61379143), the program for New Century Excellent Talents in University (NCET-11-0134), the Specialized Research Fund for the Doctoral Program of Higher Education (SRFDP) under Grant 20120161110014 and the S&T Program of Xuzhou City (XM13B119). The authors appreciate the nice help from Dr. Wang Jing of Henan Polytechnic University for her providing the source codes of exemplar-based image inpainting.

References

[1] G.K. Birajdar, V.H. Mankar, Digital image forgery detection using passive techniques: a survey, Digital Invest. 10 (3) (2013) 226–245.
[2] J.A. Redi, W. Taktak, J.L. Dugelay, Digital image forensics: a booklet for beginners, Multimedia Tools Appl. 51 (1) (2011) 133–162.
[3] O.M. Al-Qershi, B.E. Khoo, Passive detection of copy-move forgery in digital images: state-of-the-art, Forensic Sci. Int. 231 (1) (2013) 284–295.
[4] T.Y. Chang, S.C. Tai, G.S. Lin, A passive multi-purpose scheme based on periodicity analysis of CFA artifacts for image forensics, J. Vis. Commun. Image Represent. 25 (6) (2014) 1289–1298.
[5] T. Bianchi, A. Piva, Detection of nonaligned double jpeg compression based on integer periodicity maps, IEEE Trans. Inf. Forensics Secur. 7 (2) (2012) 842–848.
[6] X. Kang, M.C. Stamm, A. Peng, et al., Robust median filtering forensics using an autoregressive model, IEEE Trans. Inf. Forensics Secur. 8 (9) (2013) 1456–1468.
[7] P. Kakar, N. Sudha, Exposing postprocessed copy-paste forgeries through transform-invariant features, IEEE Trans. Inf. Forensics Secur. 7 (3) (2012) 1018–1028.
[8] Y. Huang, W. Lu, W. Sun, D. Long, Improved DCT-based detection of copy-move forgery in images, Forensic Sci. Int. 206 (1) (2011) 178–184.
[9] H.J. Lin, C.W. Wang, Y.T. Kao, Fast copy-move forgery detection, WSEAS Trans. Signal Process. 5 (5) (2009) 188–197.
[10] G. Muhammad, M. Hussain, G. Bebis, Passive copy move image forgery detection using undecimated dyadic wavelet transform, Digital Invest. 9 (1) (2012) 49–57.
[11] I. Amerini, L. Ballan, R. Caldelli, et al., Copy-move forgery detection and localization by means of robust clustering with J-linkage, Signal Process.: Image Commun. 28 (6) (2013) 659–669.
[12] S. Bayram, H.T. Sencar, N. Memon, An efficient and robust method for detecting copy-move forgery, in: Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing, 2009, pp. 1053–1056.
[13] I. Amerini, L. Ballan, R. Caldelli, et al., A sift-based forensic method for copy-move attack detection and transformation recovery, IEEE Trans. Inf. Forensics Secur. 6 (3) (2011) 1099–1110.
[14] C. Guillemot, O.L. Meur, Image inpainting: overview and recent advances, IEEE Signal Process. Mag. 31 (1) (2014) 127–144.
[15] Q. Wu, S.J. Sun, W. Zhu, et al., Detection of digital doctoring in exemplar-based inpainted images, in: Proc. of IEEE International Conference on Machine Learning and Cybernetics, vol. 3, 2008, pp. 1222–1226.
[16] K.S. Bacchuwar, K.R. Ramakrishnan, A jump patch-block match algorithm for multiple forgery detection, in: Proc. of IEEE International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013, pp. 723–728.
[17] I. Chang, J. Yu, C.C. Chang, A forgery detection algorithm for exemplar-based inpainting images using multi-region relation, Image Vis. Comput. 31 (1) (2013) 57–71.
[18] T.F. Chan, J. Shen, Nontexture inpainting by curvature-driven diffusions, J. Vis. Commun. Image Represent. 12 (4) (2001) 436–449.
[19] A. Tsai, J.A. Yezzi, A.S. Willsky, Curve evolution implementation of the Mumford–Shah functional for image segmentation, denoising, interpolation, and magnification, IEEE Trans. Image Process. 10 (8) (2001) 1169–1186.
[20] M. Jung, X. Bresson, T.F. Chan, Nonlocal Mumford–Shah regularizers for color image restoration, IEEE Trans. Image Process. 20 (6) (2011) 1583–1598.
[21] A. Criminisi, P. Prez, K. Toyama, Region filling and object removal by exemplar-based image inpainting, IEEE Trans. Image Process. 13 (9) (2004) 1200–1212.
[22] Y. Liu, V. Caselles, Exemplar-based image inpainting using multiscale graph cuts, IEEE Trans. Image Process. 22 (5) (2013) 1699–1711.
[23] J. Wang, K. Lu, D. Pan, et al., Robust object removal with an exemplar-based image inpainting approach, Neurocomputing 123 (2014) 150–155.
[24] C. Barnes, E. Shechtman, D.B. Goldman, et al., The generalized patch match correspondence algorithm, in: Proc. of European Conference on Computer Vision, 2010, pp. 29–43.
[25] G. Schaefer, M. Stich, UCID – an uncompressed colour image database, in: Proc. of SPIE, Storage and Retrieval Methods and Applications for Multimedia, vol. 5307, 2004, pp. 472–480. <http://homepages.lboro.ac.uk/cogs/datasets/ucid/ucid.html>.