Journal of Information Hiding and Multimedia Signal Processing ©2017 ISSN 2073-4212
Ubiquitous International Volume **8**, Number **1**, January **2017**

Block Sampled Matching with Region Growing for Detecting Copy-Move Forgery Duplicated Regions

Chien-Chang Chen, Ling-Ying Chen, and Yu-Jing Lin

Department of Computer Science and Information Engineering
Tamkang University
Taipei, Taiwan
ccchen34@mail.tku.edu.tw

Received May, 2016; revised July, 2016

Abstract. *A copy-move forgery region is defined as a region of an image being replaced by a copy of other region in the same image. This paper presents a novel block sampled matching with region growing algorithm (BSMRG) to detect the copy-move regions efficiently. The proposed scheme is constructed based on the assumption of the copy-move forgery region larger than a predefined region size. Test image is partitioned to non-overlapped segmented blocks according to the predefined region size. Each comparison block, which is overlapped extracted from a segmented block, is compared to the upper-left comparison block of each segmented block to find a pair of matched blocks. The copy-move forgery region can be thus acquired by applying region growing strategy to all pairs of matched blocks. Experimental results show that the proposed BSMRG can detect copy-move regions well with less computation time than other significant schemes.*
Keywords: Copy-move forgery detection, Segmented block, Somparison block, Region growing

1. **Introduction.** Digital media file has become easier than ever to modify, synthesis, and create with the rapid growth of digital devices and image/video editing software. The purpose of digital image forensics is to verify the trustworthiness of digital image/video, and it has become important in recent research topics.

Digital image forensics can be categorized to two kinds of techniques, active and passive approaches [1, 2]. Active approaches, like digital watermark techniques, proposed in the past as a way to verify the authenticity of digital images by embedding watermarks into host media [3, 4]. The embedded watermark can also detect any malicious tampering of the image. However, the active approaches have a major drawback that watermarks should be embedded into images before we use the images. This problem can be overcome by passive approaches which do not need any image information to detect the malicious tampering.

The copy-move forgery detection is a passive approach to detect copy-move forgery attack, in which a region of an image is replaced by a copy of other region in the same image. Over the past years, a large number of passive approaches for detecting image copy-move forgery modification have been proposed. These methods can be classified into categories based on DCT-based [5, 6, 7], Log-polar transform-based [8], texture and intensity-based [9], invariant key-points based [10], invariant image moments based [11], PCA-based, SVD-based, and other algorithms [12, 13, 14, 15, 16, 17, 18, 19].

For other detection algorithms, Muhammad et al. [12] adopted dyadic wavelet coefficients extracted from each block as feature vector. Lynch et al. [13] used preliminary cluster to reduce the comparison load for acquiring an efficient expanding block algorithm. Zhao et al. [14] integrated DCT and SVD techniques to extract image feature for localizing tampered regions. Chihaoui et al. [15] detected the copy-move forgery duplication by acquiring feature by the Scale Invariant Feature Transform (SIFT) method and matching by the Singular Value Decomposition (SVD) method. The proposed hybrid method is robust to geometrical transformations. Li et al. [16] segments the image into semantically independent patches prior to keypoint extraction. The copy-move duplicated regions can be detected by matching these patches. Popescu and Farid [17] used principle component analysis on exhausted blocks comparison to detect duplicated blocks and a matching algorithm is presented to acquire the duplicated regions. Recently, watermarking strategies [18, 19] are also adopted for detecting image tamper regions.

Among above works, an exhausted block matching algorithm is always required for acquiring a complete comparison between each pair of blocks. However, the exhausted block matching strategy is a time consuming method. Therefore, this paper presents a novel block sampled matching with region growing method (BSMRG) to reduce the required computation time. The proposed scheme is based on an assumption that the copy-move region is larger than a pre-defined region size. Therefore, the exhausted block matching algorithm can be then reduced to comparisons between compared blocks and sampled blocks. The test image is partitioned to non-overlapped segmented blocks and each segmented block is partitioned to overlapped compared blocks with the upper-left one denoting sampled block. The comparison required in the proposed BSMRG is between compared blocks in all segmented blocks and the only one sampled block in the remaining segmented blocks. The computation load can be thus greatly reduced.

The paper is organized as follows. Section II reviews important schemes, including the exhausted block matching algorithm and the expanding block algorithm [13], to detect copy-move forgery regions. Section III presents the proposed block sampled matching with region growing algorithm (BSMRG). Section IV presents experimental results. Section V followed by concluding remarks.

2. **Review of Important Copy-Move Forgery Detection Schemes.**

2.1. **Review of the Exhausted Block Matching Algorithm.** In this section, we briefly review the conventional exhausted block matching algorithm for detecting copy-move forgery regions. The exhausted block matching algorithm compares all possible pairs of overlapped segmented blocks in an image. Assume that the image has size of $M \times N$ and block size is defined by $k \times k$, then there are $(M-k+1) \times (N-k+1)$ overlapped blocks in the image. For ignoring neighboring blocks with partial overlapped pixels to the compared block, each block has to be compared with blocks outside the surrounding area of $(3k-2) \times (3k-2)$ and one example of the surrounding area of a block B is defined by the upper-left $(3k-2) \times (3k-2)$ block. Therefore, the searching area of the block B can be therefore defined as the whole image removing the surrounding area. For any block B, number of the possible compared blocks is $(M-k+1) \times (N-k+1) - (2k-3) \times (2k-3)$. Therefore, number of block comparison in the exhausted block matching algorithm is $\frac{(M-k+1)\times(N-k+1)\times[(M-k+1)\times(N-k+1)-(2k-3)\times(2k-3)]}{2}$. This number is enormous and worth to be efficiently reduced.

Many studies used the exhausted block matching strategy and feature extraction methods to detect copy-move forgery regions. Fridrich et al. [5] using block DCT coefficients

and Popescu and Farid [17] using principle component analysis (PCA) exhibit robust detection on copy-move forgery regions.

2.2. **Review of Expanding Block Algorithm** [13]**.** This section briefly reviews the expanding block algorithm (EB) [13] that clustering all blocks to buckets by block mean feature and applying exhausted block matching algorithm on each bucket for reducing the required computation time.

The EB algorithm uses block mean feature to reduce the number of block comparison. The algorithm first partitions the $M \times N$ image into $k \times k$ overlapped blocks to acquire $(M-k+1) \times (N-k+1)$ blocks. For improving the computation efficiency, the average intensity in each block is defined as block mean feature. Blocks are then sorted by the block mean feature and grouped uniformly into G clusters. Therefore, each cluster contains $\frac{(M-k+1)\times(N-k+1)}{G}$ blocks with closed mean features. For reducing the gap of block feature between each cluster, blocks in the neighboring cluster of ith cluster, denoted by the (i-1)th, ith, and (i+1)th clusters are combined together to construct the ith bucket. Without loss of generality, each bucket includes $\frac{3\times(M-k+1)\times(N-k+1)}{G}$ blocks. Each block is only compared with other blocks in the same bucket. Number of block comparison in the block expanding algorithm is $G \times C_2^{\frac{3\times(M-k+1)\times(N-k+1)}{G}}$. Finally, matched blocks under the same shift are combined as duplicated region.

3. **Proposed Block Sampled Matching with Region Growing Algorithm (BSMRG).** This section introduces algorithm and property discussion of the proposed scheme. Section 3.1 introduces the presented block sampled matching property. Section 3.2 introduces algorithm of the proposed block sampled matching with region growing strategy (BSMRG). Section 3.3 presents the performance improvement of the proposed scheme with other literatures.

3.1. **Block sampled matching property.** In a copy-move forgery modification problem, a region of an image called *duplicated region* is replaced by a copy of *original region* in the same image. The original region is identical to the duplicated region. Test image is partitioned to non-overlapped *segmented blocks*. *Compared blocks* are overlapped extracted from each segmented block and the upper-left compared block in each segmented block is defined as *sampled block*. This section introduces a block sampled matching property that a pair of matched blocks can be obtained by comparing all compared blocks with sampled blocks. Figs. 1(a) and 1(b) show definitions of above regions and relationships between segmented blocks and sampled blocks, respectively.

Sizes of regions and blocks are defined as follows. Assume that size of the test image is $N \times N$, size of compared block is $k \times k$, and size of the segmented block is $d \times d$. These assumptions indicate that the duplicated region should be larger than $(d+k-1) \times (d+k-1)$. For example, our experiment adopting N=256, k=16, d=32 leads to the duplicated region being larger than 47×47. The test image is partitioned to non-overlapped segmented blocks with size $d \times d$. In each segmented block, its compared blocks are defined by the block's upper-left pixel locating within the segmented block. Therefore, $d \times d$ compared blocks with size $k \times k$ can be overlapped extracted from a $d \times d$ segmented block.

The block sampled matching property is defined by that duplicated region includes at least one of the sampled block. Therefore, one $k \times k$ block can be found by comparing sampled blocks with all compared blocks. Figs. 1(c)-(f) show four examples of different matched locations between the original region and duplicated region. Figs. 1(c) and 1(d) illustrate two examples of the sampled blocks locating at the upper-left corner of the

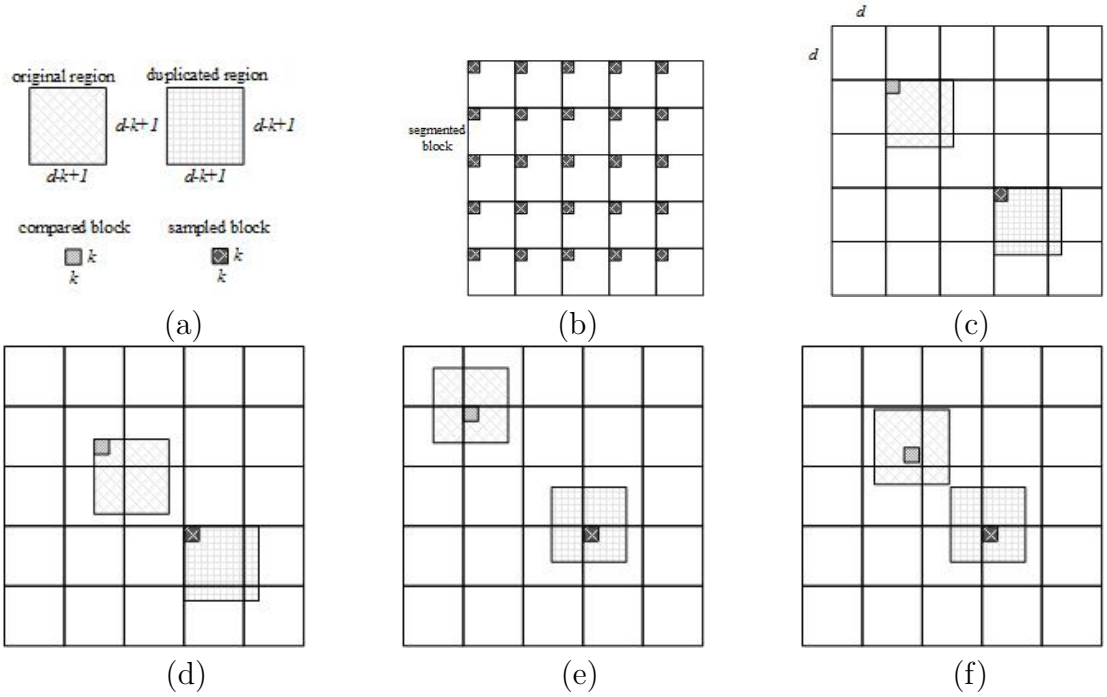


FIGURE 1. The proposed block sampled matching property, (a) region and block definition, (b) relationships between segmented blocks and sampled blocks, (c)-(f) four examples of different matched location between original and duplicated regions.

duplicated region. Figs. 1(e) and 1(f) illustrate other two examples of the sampled blocks locating at other places of the duplicated region. Fig. 1 shows that a pair of matched blocks, between original region and duplicated region, can be detected when size of the duplicated region is equal to$(d + k - 1) \times (d + k - 1)$.

3.2. **Algorithm of the proposed block sampled matching with region growing (BSMRG).** The proposed BSMRG algorithm is based on the presented block sampled matching property which is introduced in Section 3.1. By the assumption of size of the duplicated region being larger than $(d + k - 1) \times (d + k - 1)$, the duplicated region includes at least one $k \times k$ sampled block locating at a $d \times d$ segmented block. Therefore, only comparing the upper-left $k \times k$ sampled blocks in all segmented blocks to all $k \times k$ compared blocks in each segmented block can detect at least a pair of matched blocks between original and duplicated regions. The copy-move regions can be further detected by applying region growing strategy to above matched pair of blocks. The proposed BSMRG algorithm is introduced as follows.

1. Partition the $N \times N$ image to non-overlapped $d \times d$ segmented blocks to acquire $\frac{N \times N}{d \times d}$ segmented blocks.

2. The following steps are performed sequentially for each $d \times d$ segmented block.

2.1 For each $k \times k$ compared block b_c and all sampled blocks b_s, calculate the Euclidean distance dis between b_c and b_s by Eq. (1)

$$dis = \sqrt{\sum_{i=0}^{k-1} \sum_{j=0}^{k-1} (b_c(i,j) - b_s(i,j))^2} \tag{1}$$

2.2 If the calculated dis is smaller than a pre-defined threshold ETH, the block pair (b_c, b_s) is denoted by a pair of matched blocks and apply the matched pair (b_c, b_s) to Step 3 for growing the copy-move forgery regions.

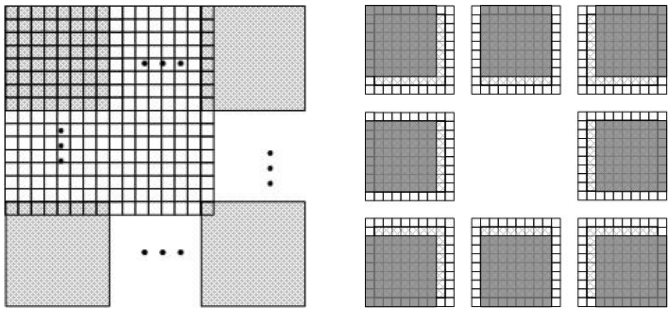
3. Assign blocks b_c and b_s to regions r_c and r_s, respectively. For regions r_c and r_s, the following steps are applied to grow the copy-move forgery regions.

3.1 For a $k \times k$ block n_c with $k-1$ pixels overlapped horizontally or vertically to the region r_c, find the block n_s with the same coordinate overlapped to the region r_s and calculate the Euclidean distance of n_c and n_s. If the Euclidean distance is smaller than the threshold ETH, add n_c and n_s to regions r_c and r_s, respectively.

3.2 Repeat Step 3.1 until no neighboring block can be added.

3.3 If the number of blocks added is larger than another pre-defined threshold DTH, then regions r_c and r_s exhibits two copy-move forgery regions.

In Step 2.1, all $k \times k$ blocks with its upper-left pixel locating within the $d \times d$ segmented block are extracted as compared blocks. Fig. 2(a) shows all extracted $k \times k$ blocks in a $d \times d$ segmented block when $k = 8$ and $d = 16$. Fig. 2(b) shows eight neighboring 8×8 blocks with 7 pixels overlapped horizontally or vertically to the central 8×8 block. In the proposed algorithm, Step 2 is the block sampled matching procedure and Step 3 is the region growing step. Moreover, based on the proposed block sampled matching property, the threshold DTH adopted in Step 3.3 is determined by the number of $(d+k-1)^2 - d^2$, in which $d \times d$ represents the segmented block size and $k \times k$ represents the compared block size. In our experiments, $d = 32$ and $k = 16$ lead to $DTH = 1185$.



(a) (b)

FIGURE 2. Example of (a) all 8×8 compared blocks extracted from a 16×16 segmented block, (b)blocks surrounded around the central 8×8 block.

4. **Experimental Results.** This section demonstrates some experimental results of our proposed scheme. All experiments were performed by MATLAB 2015 on a PC with an Intel i5-3230 CPU and 8GB RAM. Experimental results given in this section include computation time and detection rates. The detection rates are acquired after applying the test image to no attack, JPEG QF attacks, and Gaussian smooth attacks. Fig. 3 shows detected results of using two different sized segmented blocks 64×64 and 32×32 on three different test image with size 256×256. The Euclidean threshold ETH and duplication threshold DTH are empirically determined by 110 and 1185, respectively. ETH=110 is empirically determined and DTH=1185 is determined from Section 3.2.

Figs. 3(a)-(c) show three copy-move forgery images: Valley, Village, and Scenes, respectively. Detected results under two different segmented sizes, 32×32 and 64×64, are demonstrated in Fig. 4. The choice of segmented block size is to find a pair of matched blocks. Figs. 3(d)-(f) depict detected results under the assignments of 32×32 segmented

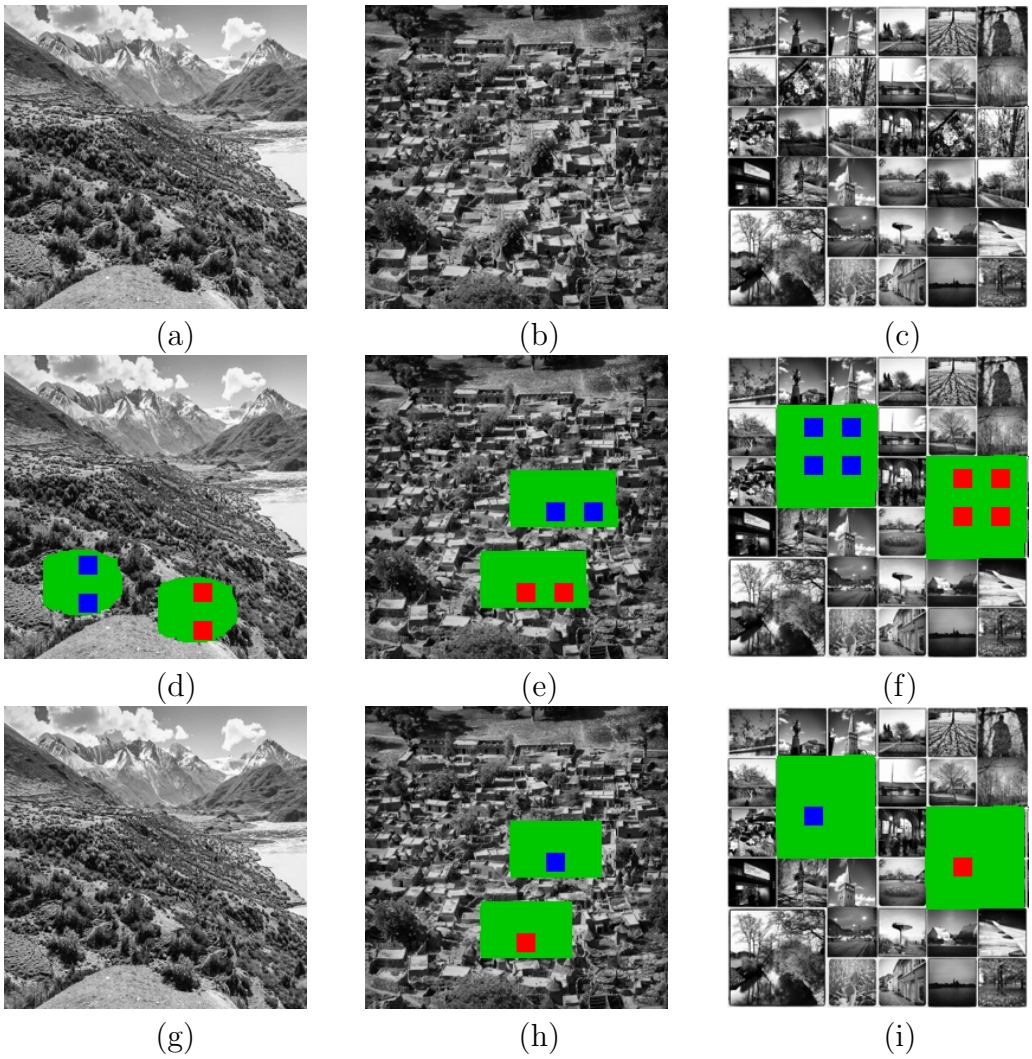


FIGURE 3. Three test images, (a)-(c) three copy-move forgery images, (d)-(f) detected results of (a)-(c) using 32×32 segmented block and 16×16 compared block, (h)-(i) detected results of (a)-(c) using 64×64 segmented block and 16×16 compared block.

block and 16×16 compared block. Figs. 3(g)-(i) depict detected results under the assignments of 64×64 segmented block and 16×16 compared block. The detected images use three different colors to represent detected regions. Each pair of red and blue squares depicts a matched pair of 16×16 compared blocks. The green areas are copy-move regions detected by the proposed BSMRG algorithm.

Comparing with the detected results between Figs. 3(d)-(f) and Figs. 3(g)-(i), we find that small segmented block size acquires more matched pair of blocks. Because an image includes more 32×32 segmented blocks than 64×64 segmented blocks, number of matched block pairs in 32×32 segmented blocks are always more than number of matched block pairs in 64×64 segmented blocks. Thus, the detected region can be better for more matched pair of blocks. Therefore, the detected results in Fig. 3(d)-(f) are always better than detected results in Fig. 3(g)-(i). Determining the size of segmented block being 32×32 or 64×64 exhibits that size of the duplicated region is at least 47×47 or 79×79, respectively.

Since the duplicated region in Fig. 3(a) is smaller than 79×79, therefore, Fig. 3(g) shows that large segmented block may fail to find a pair of matched blocks. Thus we cannot detect the copy-move forgery regions correctly. Therefore, our experiments are all performed under 32×32 segmented block and 16×16 compared block for the assumption that size of duplicated region is larger than 47×47.

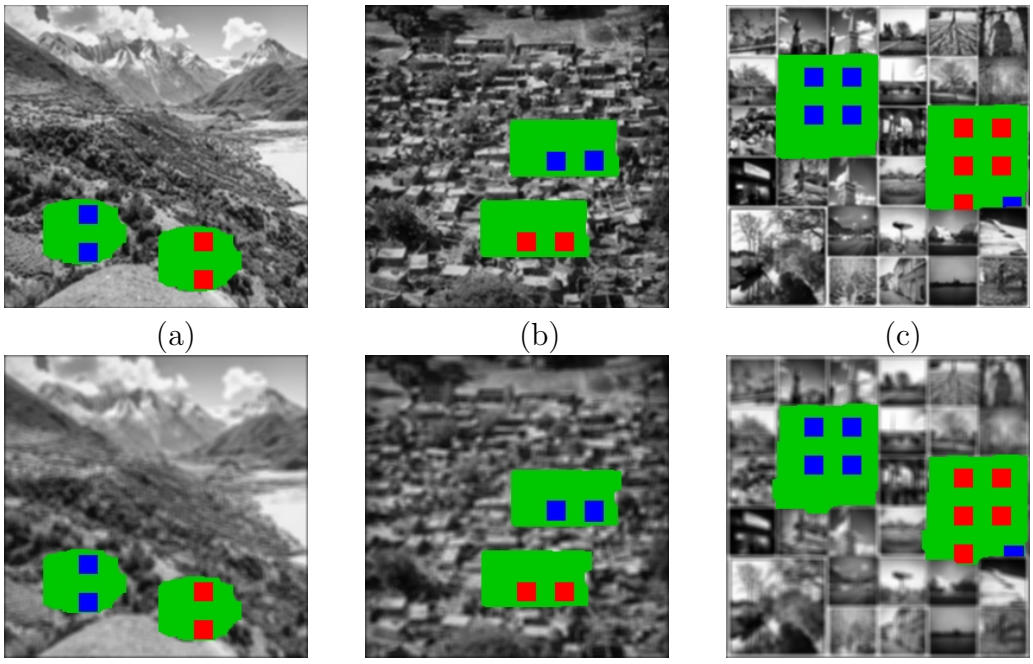
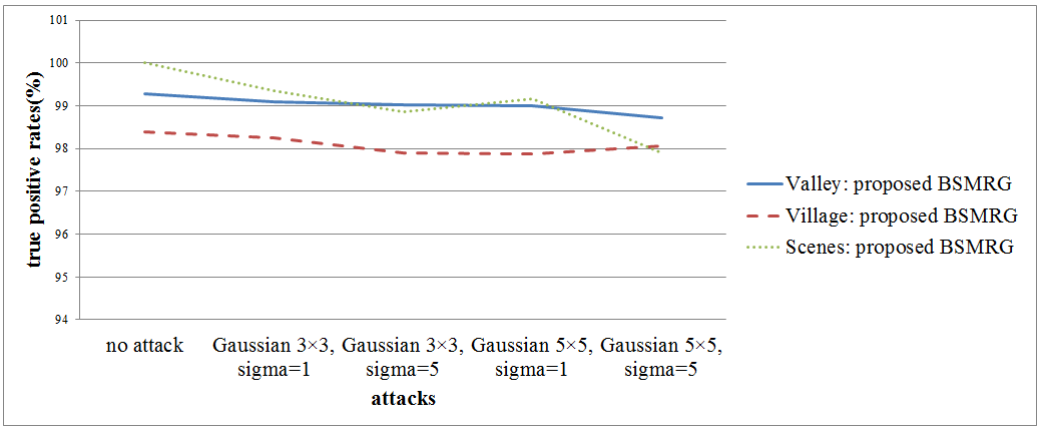


FIGURE 4. Detected results under Gaussian smoothing attacks, (a)-(c) detected results of applying three copy-move forgery images to Gaussian smoothing attack with kernel size 33 and σ=1, (d)-(f) detected results of applying three copy-move forgery images to Gaussian smoothing attack with kernel size 55 and σ=5.

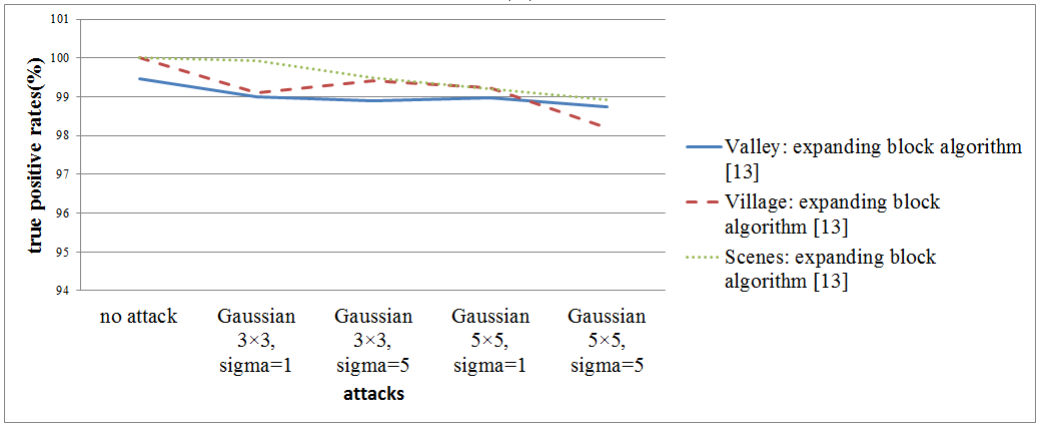
Fig. 4 show the detected results of applying Gaussian smoothing attacks to copy-move forgery images as shown in Figs. 3(a)-(c). Results of applying Gaussian smooth attacked images, as shown in Figs. 4(a)-(c) and Figs. 4(d)-(f), exhibit that the detect results of blurred copy-move forgery images are a little worse than no attack. Especially in test image Valley, the detected region is not well because the original and duplicated regions are oval, and the detected regions easily become rectangle because of the compared block being rectangle. True positive rates and false positive rates of detected results in Fig. 4 are plotted in Figs. 5 and 6.

Moreover, true positive rate and false positive rate are utilized to measure the detected results. The true positive rate is defined by $\frac{|D \cap M|}{|M|}$, where D denotes the set of pixels in detected region and M denotes the set of pixels in copy-move original region. The false positive rate is defined by $\frac{|D \cap (I-M-B)|}{|I-M-B|}$, where I denotes the set of pixels in test image and B denotes the set of pixels in copy-move duplicated region.

Fig. 5(b) and Fig. 6(b) depict all true positive rates and false positive rates in our experiments, respectively. The proposed BSMRG is performed by segmented block 32×32 and compared block 16×16. Some comparison results, acquired from the expanding block algorithm [13], are also illustrated in Figs. 5 and 6. The experimented attacks include Gaussian smooth of kernel size 33 or 55 with $\sigma = 1$ or $\sigma = 5$, and JPEG QF=95 or QF=75.



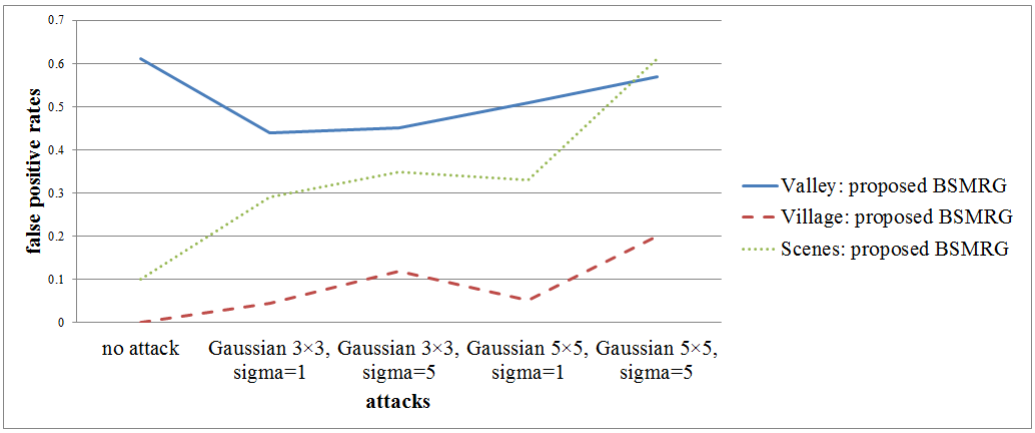
(a)



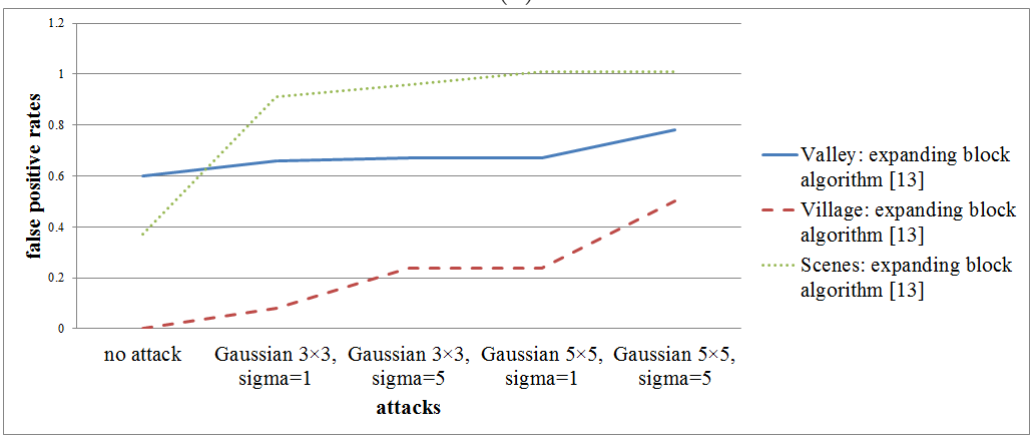
(b)

FIGURE 5. True positive rates between (a) the expanding block algorithm [13] and (b) the proposed BSMRG under different attacks.

Note that thresholds on different experiments may be also different. For example, threshold in no attack can be set to 0 for nearly perfect detection. But the threshold in JPEG QF=75 attack should be larger for giving more tolerance and our experiment adopts the threshold as 110. Fig. 5 shows that the proposed BSMRG exhibits similar true positive rates to the expanding block algorithm [13]. All true positive rates under different attacks are higher than 95%. Fig. 6 demonstrates that false positive rates are small. Moreover, false positive rates in the proposed BSMRG exhibit smaller than rates in the expanding block algorithm. Thus, we exhibit the excellence of the proposed BSMRG on acquiring good true and false positive rates. Fig. 7 depicts the computation time between the proposed BSMRG algorithm of two different assignments on segmented block being 64×64 or 32×32, the EB algorithm [13], and the exhausted block matching algorithm. Fig. 7 shows that the proposed scheme has superior performance than others. Large segmented block leads to high computation performance. The proposed BMSRG with segmented block 64×64 acquires more efficient computation time among these four methods. However, detected result in Fig. 3(j) shows that large segmented block size may not detect any matched block pair and thus fail to acquire the detected regions. Therefore, the segmented block is determined by 32×32 in our experiments and the performance is almost of the expanding block algorithm [13]. Since the classification required in expanding block algorithm and the intersection required in the proposed BSMRG can be implemented by a matrix, the spaces required in the conventional exhausted block matching algorithm,



(a)



(b)

FIGURE 6. False positive rates between (a) the expanding block algorithm [13] and (b) the proposed BSMRG under different attacks.

expanding block algorithm, and the proposed BSMRG are quite closed. Therefore, the proposed BSMRG is the most efficient algorithm among these related works.

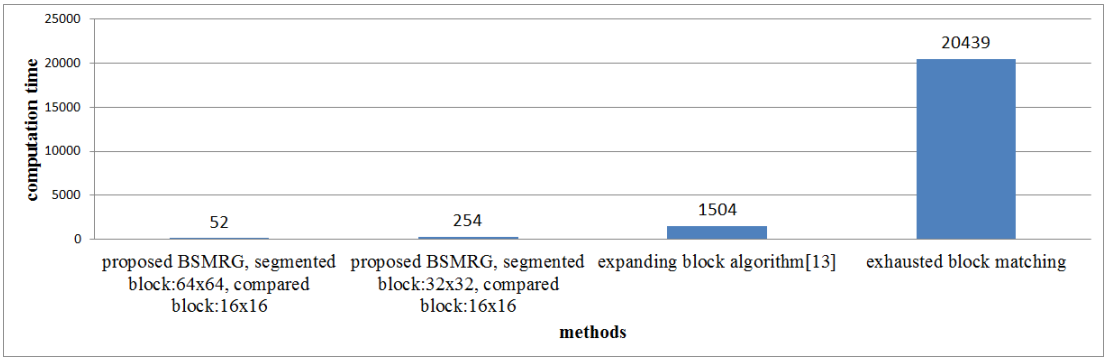


FIGURE 7. Computation time comparison between the proposed BSMRG and other methods.

Moreover, the proposed BSMRG requires limited memory complexity. The $N \times N$ image should be stored and the size is $N \times N$ bytes. Any further processings on block sampled matching and region growing can be done through the image with different indexes.

Moreover, the required Euclidean distance calculation is also space limited. Therefore, the proposed BSMRG is an efficient algorithm both in time and space complexities.

These experimental results show that the proposed scheme can efficiently detect the copy-move duplicated regions with appropriately choosing segmented block size. Comparing with the exhausted block matching algorithm and the expanding block algorithm, the proposed BSMRG exhibits best computational performance with good copy-move forgery region detection.

5. **Conclusions.** This paper presents an efficient way to detect the copy-move forgery regions in an image. In the present work, we assume that the copy-move forgery region is larger than a pre-defined size. Using the proposed BSMRG, we can efficiently detect at least a pair of matched blocks locating at the copy-move forgery region. The proposed region growing steps are then applied to generate the copy-move forgery region from the pair of blocks. Experimental results show that the proposed BSMRG can detect duplicated regions using best computation performance than the exhausted block matching algorithm and the expanding block algorithm with similar detected results. An automatic pre-processing step to predict size of the segmented block merits our future study.

Acknowledgment. This paper was partially supported by the National Science Council of the Republic of China under contract MOST 105-2221-E-032-053.

REFERENCES

[1] H. Farid, A survey of image forgery detection, *IEEE Signal Processing Magazine*, vol. 2, no. 6, pp. 16–25, 2009.

[2] O.M. Al-Qershi and B.E. Khoo, Passive detection of copy-move forgery in digital images: State-of-the-art, *Forensic Science International*, vol. 231, pp. 284–295, 2013.

[3] I. Cox, M. Miller, J. Bloom, J. Fridrich, T. Kalker, Digital Watermarking and Steganography, *Morgan Kaufmann*, 2nd edition, November 27, 2007.

[4] F. Y. Shih, *Multimedia Security: Watermarking, Steganography, and Forensics*, CRC Press, 1st edition, August 25, 2012.

[5] J. Fridrich, D. Soukal, and J. Luks, Detection of copy move forgery in digital images, in *Processings of Digital Forensic Research Workshop*, 2003, pp. 55–61.

[6] Y. Cao, T. Gao, L. Fan, and Q. Yangl, A robust detection algorithm for copy-move forgery in digital images, *Forensic Science International*, vol. 214, pp. 33–43, 2012.

[7] Y. Huang, W. Lu, W. Sun, D. Long, Improved DCT-based detection of copy-move forgery in images, *Journal of Forensic Science International*, vol. 206, pp. 178–184, 2011.

[8] S. Bravo-Solorio and A. K. Nandi, Automated detection and localisation of duplicated regions affected by reflection, rotation and scaling in image forensics, *Signal Processing*, vol. 91, no. 8, pp. 1759–1770, 2011.

[9] R. Davarzani, K. Yaghmaie, S. Mozaffari, and M. Tapak, Copy-move forgery detection using multiresolution local binary patterns, *Forensic Science International*, vol. 231, pp. 61–72, 2013.

[10] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, A SIFT-based forensic method for copy-move attack detection and transformation recovery, *IEEE Trans. on Information Forensics and Security*, vol. 6, no. 3, pp. 1099–1110, 2011.

[11] S.J. Ryu, M. Kirchner, M.J. Lee, and H.K. Lee, Rotation invariant localization of duplicated image regions based on Zernike moments, *IEEE Trans. on Information Forensics and Security*, vol. 8, no. 8, pp. 1355–1370, 2013.

[12] G. Muhammad, M. Hussain, and G. Bebis, Passive copy move image forgery detection using undecimated dyadic wavelet transform, *Digital Investigation*, vol. 9, no. 1, pp. 49–57, 2012.

[13] G. Lynch, F.Y. Shih, and H.M. Liao, An efficient expanding block algorithm for image copy-move forgery detection, *Information Sciences*, vol. 239, pp. 253–265, 2013.

[14] J. Zhao and J. Guo, Passive forensics for copy-move image forgery using a method based on DCT and SVD, *Forensic Science International*, vol. 233, pp. 158–166, 2013.

[15] T. Chihaoui, S. Bourouis, K. Hamrouni, Copy-move imageforgery detection based on SIFT descriptors and SVD-matching, *IEEE International Conference on Advanced Technologies for Signal and Image Processing*, pp. 125–129, 2014.

[16] J. Li, X.L. Li, B. Yang, and X.M. Sun, Segmentation-Based Image Copy-Move Forgery Detection Scheme, *IEEE Trans. on Information Forensics and Security*, vol. 10, no. 3, pp. 507–518, 2015.

[17] A. C. Popescu and H. Farid. Huang, Exposing Digital Forgeries bydetecting duplicated image regions, *Department of Computer Science*, TR2004-515, 2004.

[18] Z.W. Zhang, L.F. Wu, H.G. Lai, H.B. Li, and C.H. Zheng, Double Reversible Watermarking Algorithm for Image Tamper Detection, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 7, no. 3, pp. 530–542, 2016.

[19] F.C. Chang and H.C. Huang, Reversible Data Hiding with Difference Prediction and Content Characteristics, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 7, no. 3, pp. 599–609, 2016.