Digital Investigation 10 (2013) 73–84

Contents lists available at SciVerse ScienceDirect

Digital Investigation

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

Efficient image duplicated region detection model using sequential block clustering

Mohammad Akbarpour Sekeh [a,*], Mohd. Aizaini Maarof [a], Mohd. Foad Rohani [a], Babak Mahdian [b]

[a] Faculty of Computing, Universiti Teknologi Malaysia, Skudai 81300, Johor, Malaysia
[b] Department of Image Processing, Institute of Information Theory and Automation of the ASCR, Praha 8, Czech Republic

ARTICLE INFO

Article history:
Received 10 September 2012
Received in revised form 23 January 2013
Accepted 27 February 2013

Keywords:
Image forensic
Copy–paste forgery
Duplicated region detection
Local block matching
Block clustering
Time complexity

ABSTRACT

Apart from robustness and accuracy of copy–paste image forgery detection, time complexity also plays an important role to evaluate the performance of the system. In this paper, the focus point is to improve time complexity of the block-matching algorithm. Hence, a coarse-to-fine approach is applied to propose an enhanced duplicated region detection model by using sequential block clustering. Clustering minimizes the search space in block matching. This significantly improves time complexity as it eliminates several extra block-comparing operations. We determine time complexity function of the proposed algorithm to measure the performance. The experimental results and mathematical analysis demonstrate that our proposed algorithm has more improvement in time complexity when the block size is small.

1. Introduction

Existing digital image technologies provide very easy-to-use software and tools for editing digital images. Unfortunately, by these tools an expert forger can fake the image invisible to the naked eye without leaving any visual tampering clues. Digital forgeries affect social and legal systems, forensic investigations, intelligence services as well as security and surveillance systems. Image forgery detection has been created to increase the reliability of images in multimedia information systems.

Digital image forgery detection systems are designed to discover evidence of tampering by scrutinizing the forgery's clues on the image. There are several proposed methods to explore the faked image (Mahdian, 2010). Duplicated region detection is one of the more common forgery detection techniques, which identify copy–paste forgery. Copy–paste is a general type of forgeries to clone a portion of an image in order to change its semantic. It conceals an object from an image or duplicate special object. A popular workflow of copy–paste image forgery detection includes the following steps: Overlapping blocks, feature extraction, matching step and forgery detection (Fridrich, 2003).

In the duplicated region detection area, there are at least two main open problems. The first problem is finding a robust feature extraction method (Popescu and Farid, 2004; Mahdian and Saic, 2007; Pan and Siwei, 2010) and the second problem is how to improve the high computational time of the block matching step (Mahdian and Saic, 2007; Zhang, 2008; Mahdian, 2010; Akbarpour Sekeh et al., 2011; Christlein et al., 2012). For example in Mahdian and Saic (2007), the system needs at least 40 min to detect the forgeries on 640 × 480 pixels image.

The most time consuming step in the duplicated region detection is to match huge number of blocks. Computational time in matching step depends on the following metrics: image size, block size, number of blocks, feature

* Corresponding author. Tel.: +60 127960381.
E-mail addresses: moh_akb_s@yahoo.com, mohammad.akbarpour@gmail.com (M. Akbarpour Sekeh), aizaini@utm.my (Mohd.A. Maarof), foad@utm.my (Mohd.F. Rohani), mahdianb@gmail.com (B. Mahdian).

1742-2876/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.diin.2013.02.007

vector dimension, method of the feature extraction and method of the block matching (Akbarpour Sekeh et al., 2011). Based on the literature, three common approaches to improve the time complexity are decreasing the number of instances (blocks), reducing the feature vector dimension, and improving the block-matching algorithm. In duplicated region detection, the most time consuming step is to find similar blocks in the matching step. Most of the previous works (Fridrich, 2003; Popescu and Farid, 2004; Bravo-Solorio and Nandi, 2011; Huang et al., 2011; Michael Zimba, 2011) applied lexicographically based sorting as main algorithm of the matching step. This algorithm has time complexity in order of:

$$T(\alpha,\rho)\in O(\rho\alpha\ \log_2\alpha) \tag{1}$$

This complexity is a function of feature vector dimension ρ and number of blocks α.

In this paper, we focus on duplicated region detection and scrutinize copy–paste forgery clues. We concentrate on the second problem: how to reduce the time required to detect the forgeries. Here, we proposed a coarse-to-fine block-matching model using block clustering technique and local block matching. For this purpose, the sequential straightforward block clustering are applied that can enhance efficiency of the matching step by reducing the search space and grouping the similar blocks in the same clusters (Sergois and Koutroumbas, 2009). This grouping localizes the scope of the block matching into one cluster and eliminates several extra block-comparing operations.

To analyze the performance of the proposed algorithm, the time complexity function of algorithm is formulated. The experimental results and mathematical analysis demonstrate that coarse-to-fine block matching in the proposed model is more cost-effective than lexicographically-based sorting (Fridrich, 2003) when the block size is small.

This paper is organized into five major parts. In part 2, components of current workflow and research background in copy–paste image forgery detection related to improving time complexity issues are explained. In part 3, an effective Coarse-to-Fine block matching algorithm using sequential block clustering is proposed. Algorithm formulation, Sequential block clustering and local block matching will be described in some details. In part 4, time complexity function of the proposed algorithm will be formulated. In this part, the performance of proposed method is analyzed. Finally, the conclusion and future work are stated in part 5.

2. Copy–paste image forgery detection

2.1. Common workflow of duplicated region detection

Duplicated region detection is a forgery detection technique that indicates copy–paste forgery on the image. Copy–paste forgery is one of the most popular ways to change the image information semantics by cloning a portion or portions of an image into another place within the same image. This leads to changes in the semantic of image in two cases: concealing an object within the image or duplicating specific objects.

Copy–paste forgery brings into the image several near-duplicated image regions. It is important to note that duplicated regions are, for the most part, not exactly alike. This is because a skilled forger usually modifies the copied regions by applying some extra editing operations such as rotation, noising, compression, scaling, and blurring.

A common workflow of this type of forgery detection has been proposed in Fridrich (2003) and many researchers still prefer to use this workflow. Fig. 1 shows the components of this workflow.

Referring to Fig. 1, the workflow is divided into 4 major steps: overlapping blocks, feature extraction, matching step and forgery decision.

Overlapping Blocks: In the first step of copy–paste image forgery detection, the image is divided into several overlapping blocks of size $b \times b$.

Feature Extraction: Block feature is defined as a function of one or more measurements that specify some quantifiable properties of each block. The result of this step is a matrix of feature vectors. Each row of this matrix saves a block feature vector. There are several types of feature extraction methods: frequency domain, transform-based, spatial domain, statistical, histogram and color, texture, edge (Nixon and Aguado, 2008).

Block Matching: All elements in the feature vectors matrix should be sorted to find every similar blocks. Since block similarity detection in huge number of blocks requires high computational time and computational complexity is often content dependent, improving time complexity is an open problem (Zhang, 2008; Mahdian, 2010; Akbarpour Sekeh et al., 2011).

Forgery Decision: Not all matched blocks signify a forged region on the image. Therefore, another step, namely forgery decision, is required in order to reanalyze matched

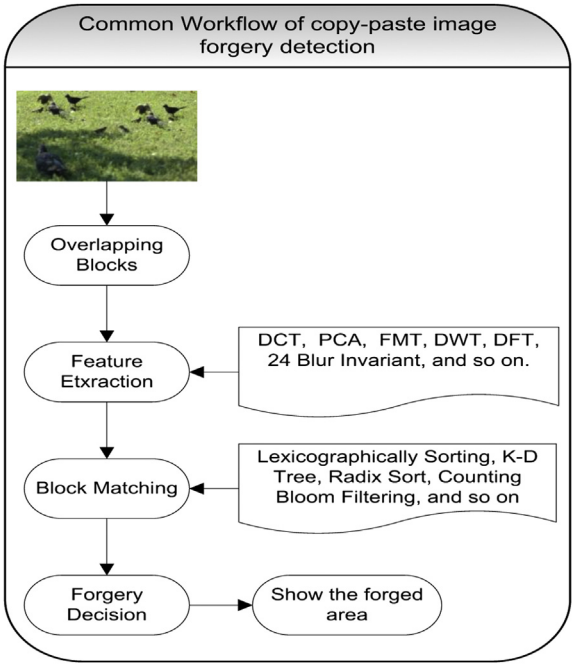

Fig. 1. Common workflow for copy–paste image forgery detection.

blocks to uncover the exact forgeries after the block-matching step. Since the duplicated region may include many overlapping blocks, the last step is to find the exact duplicated regions.

The matched blocks inside the duplicated regions have same distance. If the matched blocks are connected to each other and have same distance, they form two regions of the same shape (Jing and Shao, 2012). Therefore, the forgery decision can be made only if there are more than a certain number of similar blocks with the same distance.

2.2. Related works

Methods for improving time complexity in duplicated region detection can be categorized into at least three major approaches namely decreasing number of instances, reducing feature vector dimension and improving block-matching algorithm.

2.2.1. Decreasing the number of instances

Pan and Siwei (2010) proposed a new region duplication detection method that is robust to distortion of the duplicated regions. They estimated the transform between matched SIFT (Scale Invariant Feature Transform) key-points. In their work, only SIFT keypoints matching is performed in order to improve the time complexity. However, working with SIFT feature has several false negatives when detecting small size tampered regions. The main drawback of SIFT feature extraction compared with other image descriptors is its high computational cost (Ledwich and Williams, 2004; Blanco et al., 2010), the authors did not determine the exact time complexity function of their algorithm for further time evaluation.

2.2.2. Reducing feature vector dimension

As the feature vector dimension ρ affects the time complexity of the system, Popescu and Farid (2004) presented a method similar to Fridrich (2003). They reduced the feature vector dimension by applying Principal Component Analysis (PCA) instead of Discrete Cosine Transform (DCT). The feature vector dimension was reduced to 32.

Huang et al. (2011) developed an improved DCT-based method to detect forgery. They employed truncating to reduce the feature dimensions for use in lexicographically based Workflow and improve the time complexity of the system.

Weiqi Luo (2006) proposed a spatial domain feature for each block. They computed seven characteristic features including three features with averages of RGB channels and four features computed from color feature percentages.

Yang and Huang (2009) applied Singular Value Decomposition (SVD) to reduce the feature vector dimension. Their method transforms the image into gray-scale and lower-resolution 128×128 in order to make image smaller and to improve time complexity.

Bravo-Solorio and Nandi (2011) mapped the overlapping blocks into log-polar coordinates to compute the 1-D descriptor. They also reduced the feature vector dimension in order to improve computational cost.

Michael Zimba (2011) applied an improved Discrete Wavelet Transform and Principal Component Analysis Eigenvalue Decomposition to detect cloning forgery. They also matched the overlapping blocks with lexicographical sorting. In order to improve time complexity, their work reduced the feature dimension to eight.

2.2.3. Improving block matching algorithm

The first publication in copy–move forgery detection area was proposed by Fridrich (2003). They divided the image into overlapping blocks and extracted the feature for each block. Their work proposed DCT coefficients as block presentation. The detection of the duplicated regions was based on matching the quantized lexicographically sorted discrete cosine transform coefficients of the blocks. The DCT feature vector dimension was 64. They applied lexicographical sorting (Wiedermann, 1981) instead of an exhaustive search in the block matching step. This algorithm has complexity in order of $O(\rho\alpha\log_2\alpha)$.

Bayram (2009) proposed Fourier Mellin Transform (FMT) to enhance the robustness against scaling and rotation. They also proposed a new method namely Counting Bloom Filters with hashing the feature vectors to improve the efficiency and reduce the time complexity. However, finding an effective hash function is not easy. This technique also affects the robustness of the system.

Mahdian and Saic (2007) proposed a method for detecting near duplicated regions. In order to enhance the robustness against blurring, Moment invariants and PCA were applied. They reduced computational time by applying hierarchical structure kd-tree. However, Moment feature and PCA will raise time complexity in the feature extraction step. Moreover, the complexity of kd-tree depends on the distribution of similar intensity blocks. Hence, high computational time in their work is still an open-problem (Zhang, 2008; Mahdian, 2010).

Zhang et al. (2010) proposed a method based on Scale Invariant Feature Transform to detect the forgeries. They also used Efficient Subwindow Search (ESS) algorithm for improving the time complexity. Their work reduced the complexity to $O(kP)$. In which k is the number of key-points and P is total number of pixels. However, they did not mention how to improve the time complexity in details, because the time complexity of ESS is at least in order of $O(n^3)$ (Senjian et al., 2009).

Lin et al. (2009) proposed a new block feature extraction method in order to improve robustness against compression and noising. They represented each overlapping block by 9-dimensional feature vector in spatial domain. They applied efficient Radix-sort for performing the lexicographical sorting with order of $O(n(256+k))$ where k is the number of blocks and n is the feature vector dimension. However, Radix-sort limits the type of feature vector elements to only *integer* value and cannot always be used with other different feature vectors.

3. Coarse-to-fine block matching algorithm

The main reason behind the high computational time in lexicographically based block matching algorithm (Fridrich, 2003) is its blind similarity searching. This non-intelligent

method performs several extra block-matching operations. For example in a nature photo including areas of sky, grass and sea, lexicographical sorting blindly compares all features extracted from overlapping blocks in the image, without considering the type of region (Differences in texture and luminance). However, there is no need to compare overlapping blocks extracted from the sea region with the blocks from areas of the grass. The question is how to impart some intelligence to a block-matching algorithm in order to reduce this extra block matching?

In this paper, we improve the structure of matching algorithm and propose an enhanced model based on a coarse-to-fine approach using block-clustering technique. In this model all similar or forged blocks will be grouped in the same clusters (Sergois and Koutroumbas, 2009). Hence, for detecting forged regions it is enough to compare the blocks of each group separately. Therefore, using block clustering prior to the exact high dimensional feature feature for mathematically performance analysis. Algorithm formulation for proposed model and explanation of these two features are mentioned in the next sections.

3.1. Algorithm formulation

Fig. 2 shows the proposed duplicated region detection model. As shown in Fig. 2, there are two block-matching components in the new model: Coarse-match (Sequential block clustering) and Fine-match (Local block matching). As mentioned earlier, two types of features are needed to do these matches which known as low accurate feature and high accurate feature. Low accurate feature is proposed as a criterion to find similar cluster in the block-clustering step and High accurate feature is used to detect matched blocks (forged regions) in the local block-matching step. Based on this model, the proposed two-layer block matching algorithm is designed by the following pseudocode:

Pseudocode for Proposed Coarse-to-Fine Block Matching Algorithm:

```
1: img = Getimage(image);
2: MatchedB = Ø ;
3: OverlappingBlocks = Divide(img);
4: CreateEmptyBST(Clusters);  % BST structure for clusters
5: FOREACH (OverlappingBlocks as CurrentBlock)
6: {
7:   L = LowAccurateFeature(CurrentBlock);
8:   H = HighAccurateFeature(CurrentBlock);
9:   IF (C_k= FindSimilarCluster(L)) THEN        %Coarse block matching
10:  {
11:    LocalMatch= LocalBlockMatching(H, C_k.elements);  % Fine block matching
       MatchedB = MatchedB ∪ LocalMatch;
12:    Insert(H, C_k);
13:  }
14:  ELSE
15:  {
16:    CreateNewEmptyCluster(C, L);  % BST structure for saving H Features
17:    Insert(H, C);
18:    Insert(C, Clusters);
19:  }
20:  ENDIF
21: }
```

Declaration of Variables in above Pseudocode

Img: Input image	**MatchedB**: Matched blocks
OverlappingBlocks: Ovelapping blocks	**Clusters**: a BST structure for saving the clusters
CurrentBlock: Current block pixels (bxb)	**L**: LOW accurate feature of current block
H: High accurate feature of Current block	**C_k** : *A* cluster similar to Low feature of current block
LocalMatch: Result of Local block matching	**C_k.elements**: Each element is a High accurate feature vector

matching will reduce the search space and the number of matching operations. This reduction will significantly improve time complexity.

The matching process in proposed coarse-to-fine matching method is performed by two-layer block matching with two types of features. The first match (coarse-match) is for clustering the blocks by matching the low accurate features, while the second match (fine-match) is to find the exact similar blocks by matching the high accurate features. Since the focus point in this research is on improving the time complexity without knowing about the type of the features, we do not perform the feature robustness evaluation. The features will be applied in this paper as black-box. And only feature vectors $\overrightarrow{L}$ and $\overrightarrow{H}$ with dimension ρ_1 and ρ_2 are used as low and high accurate

The algorithm is divided into two parts. The first part includes obtaining a new image and putting it into img (Line 1), initializing the matched block array MatchedB (Line 2), dividing the image into several overlapping blocks (Line 3) and creating an empty binary search tree (BST) (William Ford, 2005) for saving the clusters (Line 4).

In the second part, all overlapping blocks are processed one by one. In lines 7–8, Low and High accurate features of current block are extracted. Sequential block clustering (Coarse-match) then tries to find a cluster similar to the current block (Line 9). In this line, Low accurate feature is a criterion to find a similar group. If a similar group is found, the local block matching (Fine-match) will be executed according to High accurate feature (Line 11). On the other hand, if the grouping algorithm could not find any similar

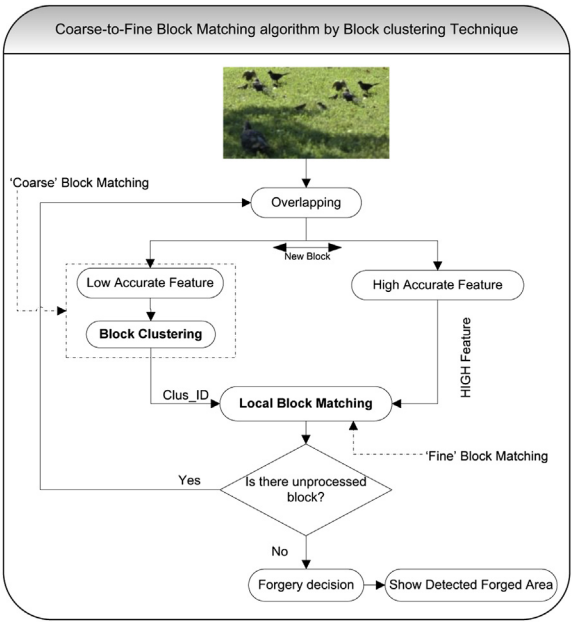

Fig. 2. Proposed two layer block matching model.

cluster based on similarity threshold, a new cluster will be created to save all coming blocks similar to the current block (Line 16–18). In this case, local block matching for this block is not executed. These instructions should be executed for all α blocks. All matched blocks are saved in the match block array *MatchedB*. This array is an output of this algorithm and will be used for the final decision in the forgery detection step.

The following sections explain sequential block clustering (Coarse-match) and local block matching (Fine-match) in detail.

3.2. Sequential block clustering

In the proposed model, coarse-match (block clustering) is defined as a new component in image forgery detection. We propose a straightforward sequential block-clustering algorithm for classifying the blocks into clusters in order to localize the block matching.

Block clustering is a technique used to group similar blocks wherein blocks of one cluster should be similar to one other and dissimilar to the blocks of other clusters. The output of the clustering process is a set of clusters where each block is uniquely assigned to a single cluster. Straightforward sequential algorithm is the fastest clustering technique that can be used to classify a set of data points into clusters based on a distance measure. In this type of clustering, all the feature vectors are presented to the algorithm once. In this case, the number of clusters is not known at first. In fact, new clusters are created as the algorithm evolves. If there is no similar cluster with new block, algorithm creates a new cluster and puts the block in it (Sergois and Koutroumbas, 2009).

The updated version of sequential clustering algorithm (Sergois and Koutroumbas, 2009) is stated as:

Updated pseudocode of Sequential Block Clustering:

$\psi = 1$;
$C_\psi = L_1$;
FOR $i = 2$ *TO* α
  Find C_k: $d(L_i, C_k) < \theta$);
  IF (C_k *was found*) THEN
    $C_k = C_k \cup L_i$;
  ELSE
    $\psi = \psi + 1$;
    $C_\psi = L_i$;
  End (IF)
End (FOR)

Declaration of variables in above pseudocode:

ψ: number of cluster
L: Low accurate block Feature vecotor
C_k: cluster k where the distance of L and characteristic of this cluster is less than θ

Note: If the new L (low accurate feature vector) is similar to C_k then this vector should be joined to this cluster. Otherwise, if the new L was not assigned to any cluster, a new cluster will be created. The characteristic of this cluster is initialized with low accurate feature vector of this block.

Let $d(L, C)$ denote the distance between a low accurate feature vector $\overrightarrow{L}$ and a cluster C. The user-defined parameter required by the algorithm is the threshold of dissimilarity, θ. Let ψ be the number of clusters that the algorithm has created and α be the number of blocks.

The important part of this algorithm is the measurement of the distance between $\overrightarrow{L}$ and C which is denoted by $d(\overrightarrow{L}, C)$. In proposed method, low accurate feature ($\overrightarrow{L}$) is extracted from each block as a similarity criterion for measuring $d(\overrightarrow{L}, C)$. The distance $d(\overrightarrow{L1}, C1)$ is Euclidean distance (Elena Deza, 2009):

$$d\left(\overrightarrow{L1}, C1\right) = \sqrt{\sum_{i=1}^{\rho_1} (L1_i - C1_i)^2} \tag{2}$$

In this equation, cluster 1 is denoted with $C1$. The structure of each cluster has at least two main fields: one field for saving the main characteristic of the cluster and second field is for saving the robust feature vector of the blocks that their low accurate features are similar to the cluster characteristic. Here the cluster characteristic is initialized with the low accurate feature of the first block when the cluster is created. Low accurate feature $\overrightarrow{L1}$ is the feature vector of block 1 with dimension ρ_1 as follow:

$$\overrightarrow{L1} = (L1_1, L1_2, L1_3, L1_4, \ldots, L1_{\rho 1}) \tag{3}$$

To choose a suitable low accurate feature (Low), note that:

- Time complexity of Low feature extraction should be reasonable.
- Dimension of Low accurate feature (ρ_1) affects the time complexity of the block clustering algorithm.
- A Low feature would be more effective if it is a more discriminative feature leading to creation of more image block groups.
- Dimension of Low feature should be much less than high accurate feature.
- The small block size do not bring a considerable false clustering and only affect the number of clusters, because low accurate feature of the blocks are rough and are used only to reduce the search space in Second match.

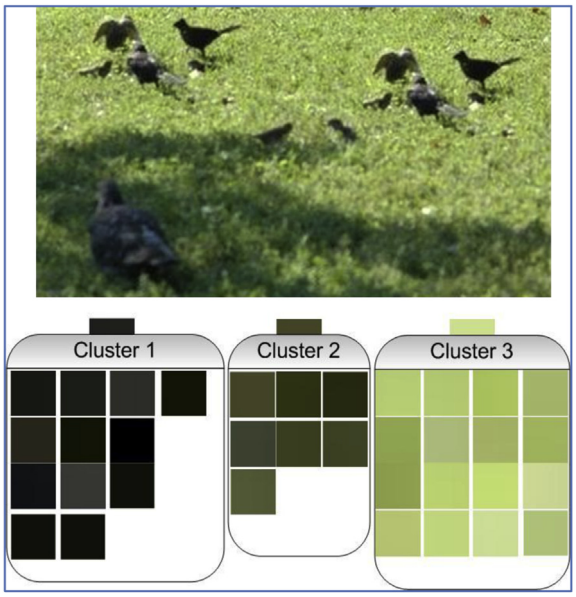

Fig. 3. In the coarse-match step, all the image blocks are clustered based on low accurate feature.

Consequently, Low-order image block statistics can be used as Low feature in our model. These statistical features may include spatial frequency feature (Li and Wang, 2005), color intensity-mean, color saturation-mean, color hue-mean (Xiaoyuan et al., 2004), density, contrast, histogram (Nixon and Aguado, 2008) and variance, skewness, kurtosis (Pitas, 2000).

3.3. Local block matching

In the proposed model, local block matching (Second match) is executed after block clustering. High accurate feature of each block is used in this step to find similar blocks in each cluster. Local block matching means the blocks of each cluster are only compared with the blocks of their clusters. Hence, the amount of extra block matching is reduced. Here if two blocks are similar each other, the high accurate feature of these blocks will be very close (Fig. 4).

Here, a binary search algorithm (Richard Neapolitan, 2011) with distance measure (Elena Deza, 2009) is an effective method to find the similar blocks because the blocks of each clusters have been saved in BST tree (William Ford, 2005). In this step, High accurate feature of blocks $\overrightarrow{H}$ should be compared which is a vector with dimension ρ_2,

$$\overrightarrow{H} = (h_1, h_2, h_3, h_4, \ldots, h_{\rho_2}) \tag{4}$$

The Euclidean distance (Elena Deza, 2009) of two High accurate feature is determined as:

$$d\left(\overrightarrow{H1}, \overrightarrow{H2}\right) = \sqrt{\sum_{i=1}^{\rho_2} (H1_i - H2_i)^2}$$

Dimension of High accurate feature affects the time complexity of the local block-matching step. Best High feature would be a block feature with minimum feature

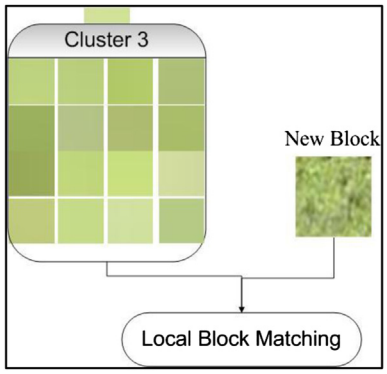

Fig. 4. In the Fine-Match step, high accurate feature of the new block will be compared with the high accurate features of the blocks in the same cluster. This step is called local block matching.

vector dimension and high robustness. Computational time for extracting this feature also affects the total time complexity of the system. The authors in Mahdian and Saic (2007), Bayram (2009), Pan and Siwei (2010), Michael Zimba (2011) proposed several robust feature extraction methods. These features can be applied as High feature in local block matching (Second match).

For reducing the false alarm and extra block matching, there is no need to compare two blocks that are so close. Hence, there is a need for calculating the distance of the two blocks. If the distance of two overlapping blocks is more than threshold, it means two blocks are not so close. The distance of two blocks B1 and B2 can be calculated by:

$$Distance(B1, B2) = \sqrt{(x_i - x_j)^2 + (y_i - y_i)^2} \tag{5}$$

Time complexity of local block matching is in order of $O(\rho_2 \beta log_2 \beta)$ in which ρ_2 is high accurate feature vector dimension and β denotes average number of blocks in each cluster. The result of this step is denoted by *Matchedblock* to identify matched blocks. This vector should be presented into the forgery decision step in order to find exact duplicated regions.

4. Results and performance analysis

When analyzing the efficiency of an algorithm in terms of time, we do not determine the actual number of CPU cycles because it depends on the particular computer on which the algorithm is running. Furthermore, we do not even want to count every instruction executed, because the number of instructions depends on the programming language being used and the way the programmer writes the program. Rather, we want a measure that is independent of the computer, the programming language, and the programmer. Here, time complexity analysis of an algorithm can determine how many times the basic operation is done for each value of the input size (Richard Neapolitan, 2011).

4.1. Time complexity

Time complexity of the proposed algorithm should be specified in order to analyze the performance. Therefore,

time complexity of each of the components in our algorithm is determined as follows:

1. *Overlapping Block*: With block size $b \times b$ and image size $M \times N$ the image should be divided into α overlapping blocks:

$$\alpha = (M - b + 1) \times (N - b + 1) \tag{6}$$

The basic operation in this part is specifying the next overlapping block. There are α passes through the loop. Therefore, the basic operation is always done α times. Hence, the complexity of this part is in order of:

$$T_1(\alpha) \in O(\alpha) \tag{7}$$

2. *Feature Extraction*: The feature of each block in the best-case can be extracted by one time passing in $O(b^2)$. The basic operation in this step is traversing the blocks pixel-by-pixel. There are α passes through the loop because the number of blocks is α. Therefore, total time complexity of both Low and High accurate feature is in order of:

$$T_2(\alpha, b) = \alpha b^2 + \alpha b^2 \in O(2\alpha b^2) \tag{8}$$

3. *First-Match for clustering the blocks*: The basic operation in this step is sequential block clustering which is done by comparing Euclidean distance between the low accurate feature of block and the clusters feature to find a similar cluster. By applying Binary Search in this step, the maximum number of comparing operations is $\log_2\psi$ (William Ford, 2005). Therefore, the complexity function for all α blocks can be estimated as:

$$T_3(\alpha, \psi, \rho_1) = \frac{\psi \text{ terms}}{\rho_1(\log_2(1) + \log_2(2) + \cdots + \log_2(\psi)} + \frac{(\alpha - \psi)\text{terms}}{\log_2(\psi) + \log_2(\psi) + \cdots + \log_2(\psi))} = \rho_1(\log_2(\psi!) + (\alpha - \psi)\log_2(\psi)) \tag{9}$$

where,α is number of terms or number of blocks, ψ is number of clusters and ρ_1 is LOW feature vector dimension.

After simplifying equation (9) with Stirling's approximation (Abramowitz and Stegun, 1972), the complexity of First-match will be in the order of:

$$T_3(\alpha, \psi, \rho_1) \in O(\rho_1 \alpha \ \log_2(\psi)) \tag{10}$$

4. *Local Block Matching (Second-Match)*: The basic operation in second-match is comparing Euclidean distance between the high accurate features of the current block with the previous blocks in the same cluster to find matched blocks. We applied Binary Search in this step. Hence, the maximum number of comparing operations in this step is $\log_2\beta$. Therefore, the complexity function for all ψ clusters can be formulated as:

$$T_4(\alpha, \psi, \rho_2) = \rho_2\psi \frac{\beta \text{ terms}}{(\log_2(1) + \log_2(2) + \cdots + \log_2(\beta))} \tag{11}$$

where,

$\beta \cong \alpha/\psi$ is the average number of blocks in each cluster and ρ_2 is HIGH feature vector dimension.

After simplifying equation (11) with Stirling's approximation (Abramowitz and Stegun, 1972) we have:

$$T_4(\alpha, \psi, \rho_2) \in O(\rho_2\psi \ \log_2(\beta!)) \in O(\rho_2\alpha \ \log_2(\beta)) \tag{12}$$

where $\psi \cong \alpha/\beta$.

The total time complexity T is determined by the summation of four complexity functions as:

$$T(\alpha, b, \psi, \rho_1, \rho_2) = T_1(\alpha) + T_2(\alpha, b) + T_3(\alpha, \psi, \rho_1) + T_4(\alpha, \psi, \rho_2)$$

So, the time complexity of clustering based local block matching algorithm is denoted as:

$$T(\alpha, b, \psi, \rho_1, \rho_2) \in O(\alpha + 2\alpha b^2 + \rho_1\alpha \ \log_2(\psi) + \rho_2\alpha \ \log_2(\beta)) \tag{13}$$

where:

$\rho_1 < \rho_2$, ρ_1 should be much less than ρ_2,
$\beta \ll \alpha$, β is much less than α.

The most important parameter which affects the performance of the proposed method is the number of clusters ψ. Therefore, equation (13) can be divided into the following three cases by value of ψ:

$$T(\alpha, b, \psi, \rho_1, \rho_2) \in \begin{cases} O(\alpha + 2\alpha b^2 + \rho_1\alpha\log_2(\alpha)) & \psi = \alpha, \\ O(\alpha + 2\alpha b^2 + \rho_1\alpha\log_2(\psi) + \rho_2\alpha\log_2(\beta)) & 1 < \psi < \alpha, \\ O(\alpha + 2\alpha b^2 + \rho_2\alpha\log_2(\alpha)) & \psi = 1. \end{cases} \tag{14}$$

The first case is $\psi = \alpha$ which occurs when the most ideal and suitable low accurate feature on the high complex image has been chosen. In this case, due to the high discriminative level of the feature, the number of clusters will be equal to the number of blocks. However, in the forged image, this case is impossible because some areas have been duplicated by forger.

Another case is $\psi = 1$ which occurs when no appropriate low accurate feature has been chosen. This is the worst case of our algorithm where all blocks are grouped in one single cluster. It means that all blocks have near low accurate feature thereby they are not clustered by used features. Consequently, the time complexity of the algorithm is not going to be improved by applying the proposed method in this case.

The case $1 < \psi < \alpha$ is the normal case in which a higher number of clusters lead to more improvement of the time complexity.

4.2. Performance analysis

We analyze the performance of the proposed algorithm by comparing equation (13) with time complexity function of lexicographically sorting algorithm (equation (15)). Lexicographical sorting algorithm is a common method used in the block matching step initially proposed by Fridrich (2003).

$$T_{Lexico}(\alpha, b, \rho) \in O\left(\alpha + \alpha b^2 + \rho\alpha \log_2(\alpha) + \rho\alpha\right) \quad (15)$$

The equation (15) is a function of three main parameter which include block size b, robust feature vector dimension ρ (Equivalent to the ρ_2 in equation (13)). However, Time complexity of our algorithm (equation (13)) is a function of four main parameters: block size b, number of clusters ψ, low accurate feature vector dimension ρ_1 and high accurate feature vector dimension ρ_2. We used these parameters as metrics for comparing the performance of the proposed algorithm with lexicographically based method (Fig. 5). Therefore, we draw the time growth charts for equations (13) and (15) as shown in Fig. 5. The time complexity of two algorithms are compared based on the following variables for an image with a dimension of 648 × 480:

1 - *Block size b*: Fig. 5(a) shows a time complexity chart by changing the value of block size. The chart was created in the following situation: number of cluster ψ is assumed to be 50, $\rho_1 = 2$ and $\rho_2 = 20$. The result shows that block size directly affects the time

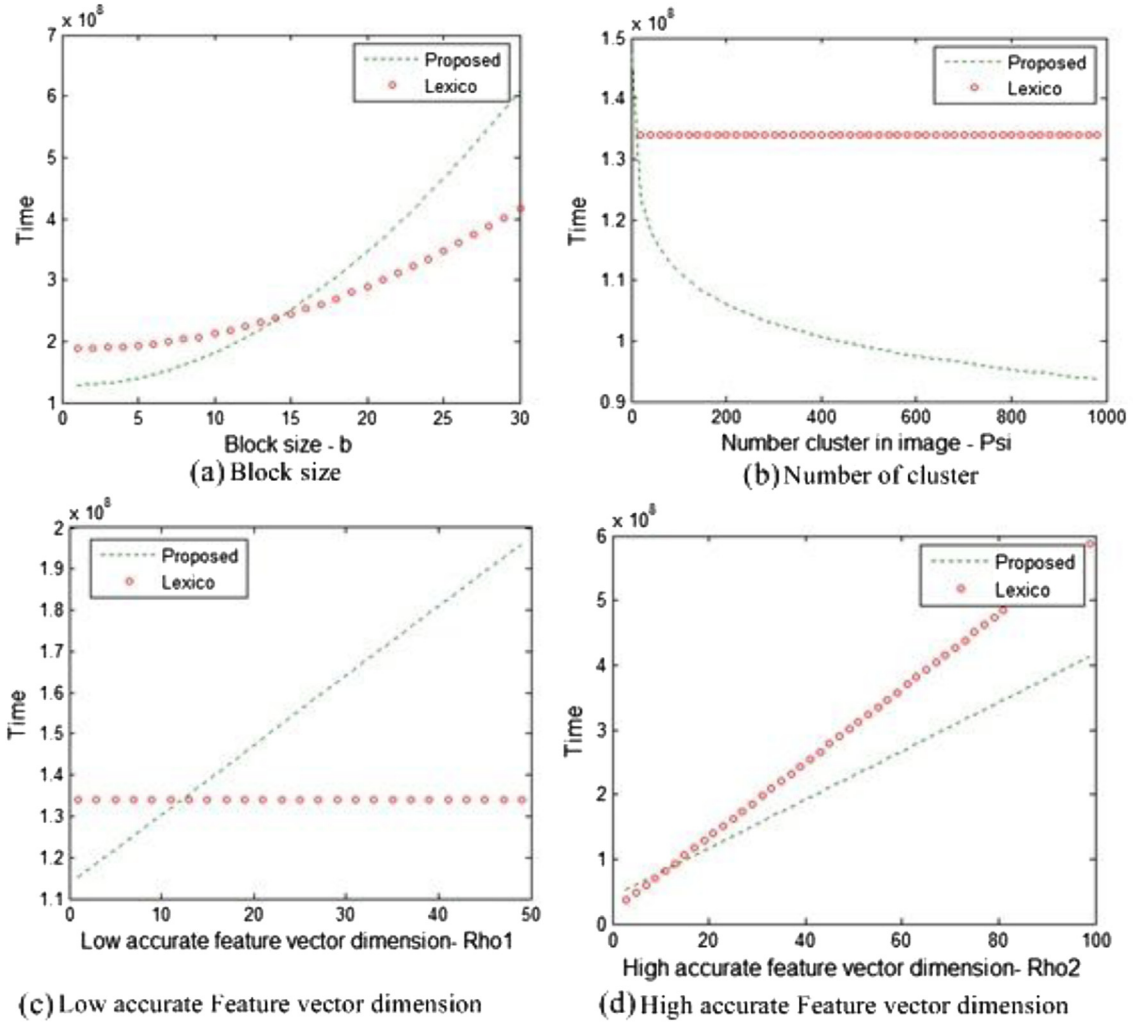

Fig. 5. Comparing time growth chart of the proposed method with lexicographically sorting.

complexity. As can be seen in Fig. 3(a), when the block size is smaller, our method is more efficient than lexicographical sorting.

2 - *Number of clusters* ψ: Fig. 5(b) shows the time complexity functions according to the changing the number of clusters. We assumed that $b = 8 \times 8$, $\rho_1 = 2$ and $\rho_2 = 20$. Fig. 3(b) shows that when the number of clusters is increased, computational time would be reduced significantly in the proposed algorithm. This Figure shows that when the number of clusters is more than threshold, computational time of our method is more efficient than Lexicographical method.

3 - *Low accurate feature vector dimension* ρ_1: Fig. 5(c) compares time growth chart of the proposed method and lexicographical sorting according to low accurate feature vector dimension ρ_1, where $\rho_2 = 20$, $b = 8$, $\psi = 50$. The result demonstrates that the proposed method is more efficient than lexicographical sorting approach when ρ_1 is less than threshold.

4 - *High accurate feature vector dimension* ρ_2: Fig. 5(d) compares the performance of two layer block matching with Lexicographical sorting according to the value of high accurate feature vector dimension ρ_2, where $\rho_1 = 2, b = 8, \psi = 50$. Referring to Fig. 5(d), it is clear that when applying longer high accurate feature, our method has better time complexity.

Threshold for number of clusters ψ and Low accurate feature vector dimension ρ_1 are determined by finding the intersection point of two curves. Referring to equations (13) and (15) to find the intersection point, we have:

$$\log_2\psi = \frac{b^2 - \rho_2}{\rho_2 - \rho_1} \tag{16}$$

After simplifying, the minimum threshold value of ψ and maximum threshold value of ρ_1 are:

$$\psi = 2^{\frac{b^2-\rho_2}{\rho_2-\rho_1}} \tag{17}$$

$$\rho_1 = \rho_2 - \frac{b^2 - \rho_2}{\log_2\psi} \tag{18}$$

So, when $\psi > 2^{\frac{b^2-\rho_2}{\rho_2-\rho_1}}$ or $\rho_1 < \rho_2 - \frac{b^2 - \rho_2}{\log_2\psi}$, the time complexity of our method is more efficient than Lexicographically-based method, as shown in Table 1.

The threshold ψ depends on block size b, low accurate feature vector dimension ρ_1 and high accurate feature vector dimension ρ_2. As Table 1 shows, the proposed method delivers greater performance improvements as block sizes become smaller and the distance between feature vectors ρ_1 and ρ_2 gets bigger.

Note that choosing smaller block size can improve the robustness and increase the ability to detect small size forgeries. However, in the lexicographically method, this will increase the number of blocks and leads to increase the time complexity.

Table 1
Threshold of ψ for efficiency of proposed method in different cases.

Case	Block size	ρ_1	ρ_2	Threshold for ψ
1	8×8	1	64	$\psi > 2^0$
2	8×8	4	16	$\psi > 2^4$
3	10×10	2	32	$\psi > 2^{2.2}$
4	24×24	1	64	$\psi > 2^{8.1}$
5	36×36	4	10	$\psi > 2^{191}$

4.3. Experimental results

The proposed local block-matching algorithm with sequential block clustering was implemented using Java language. The system platform had an Intel dual core 2.2 GHz processor and 4G RAM with a Vista operating system. The proposed method is evaluated using the

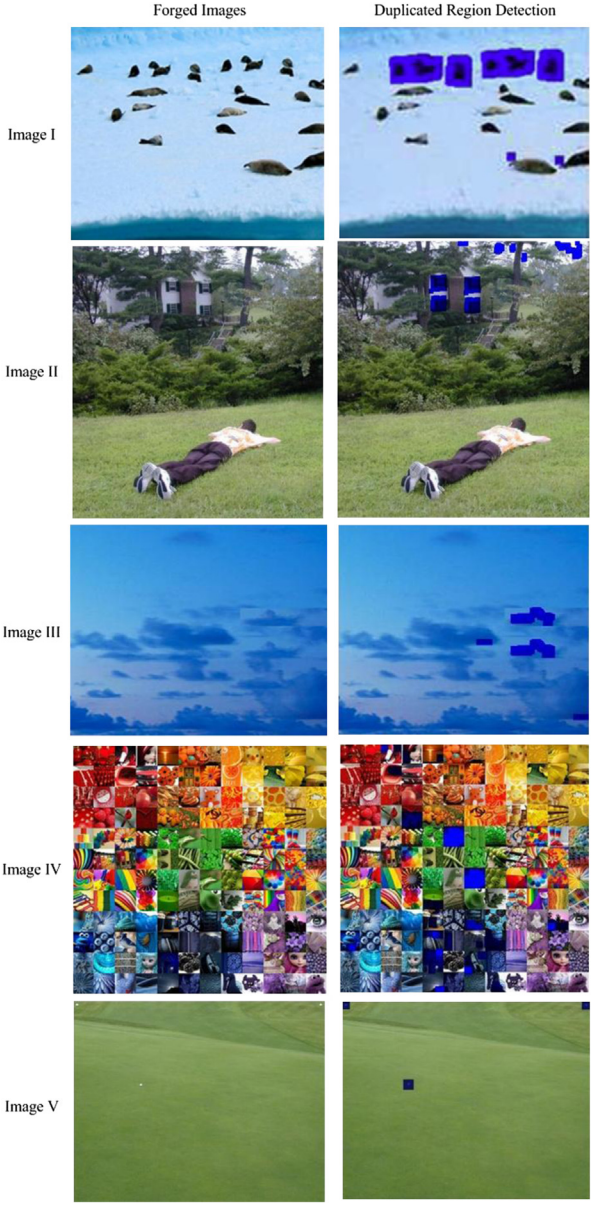

Fig. 6. Applied images from personal collection.

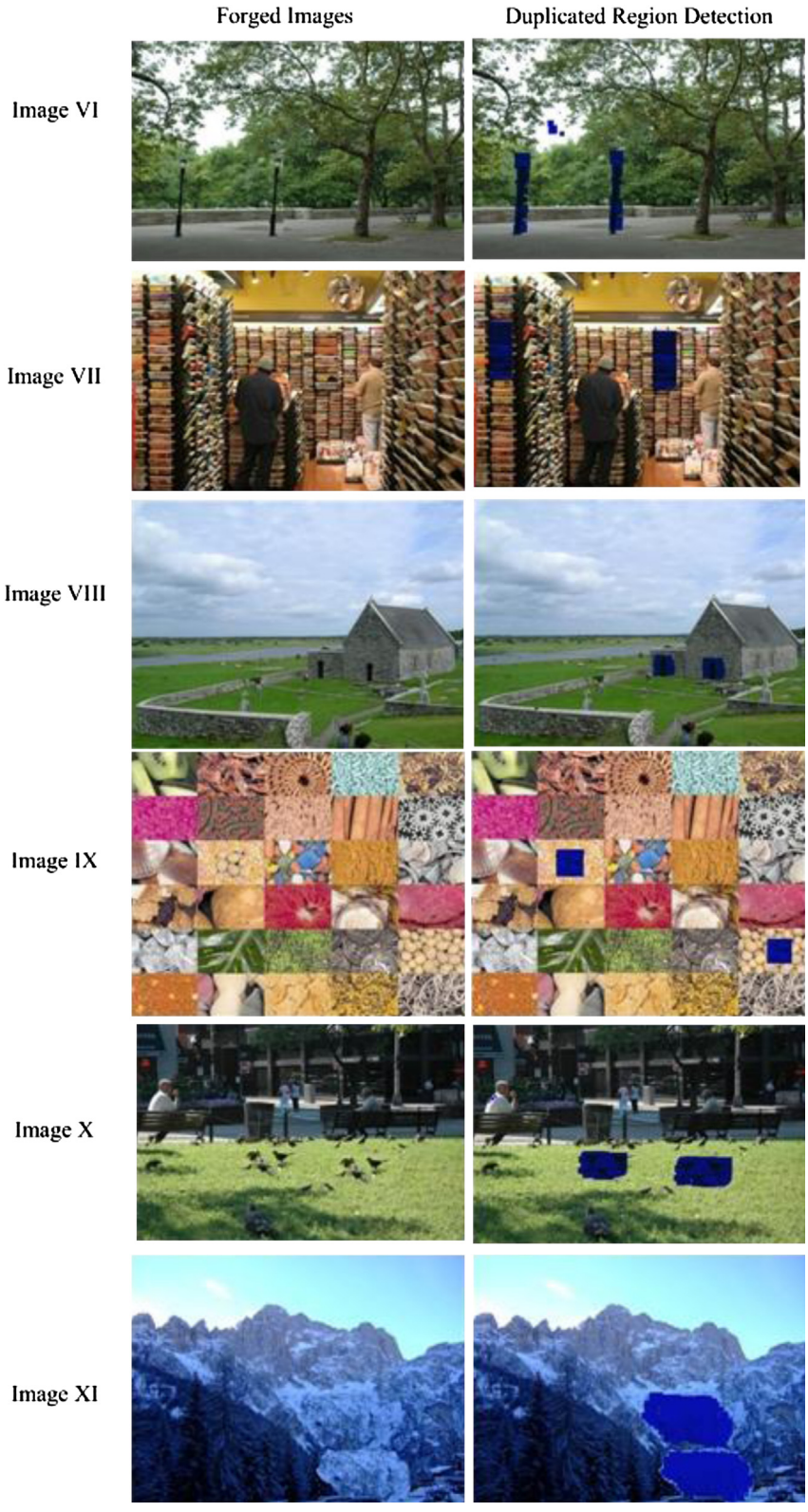

Fig. 7. Images from MICC-F220 dataset.

images from the datasets MICC-F220 and also from a personal collection. The MICC-F220 dataset consists of 221 natural images: 110 forged images and 111 originals. We also did some duplicated regions by copying a portion of the images and pasting them somewhere else in the same images in personal collection. Applied feature extraction method is DCT included 64 DCT coefficient with the block size 8×8. Spatial Frequency (Li and Wang, 2005) with dimension 1 was also applied as Low accurate feature. In fact, it is inevitable that longer block feature vectors can save more information about the blocks and bring more robustness to the matching step. The computational time of our work was compared with common Lexicographical method (Fridrich, 2003). The performance of coarse-to-fine block matching algorithm is evaluated for some cases as follow:

Image-I (Pole): Fig. 6 (the pole) with 227×191 pixels shows a photo of a pole. The proposed clustering-based method created 26 clusters. Here, the proposed algorithm had a 26.7% performance improvement. The computational time has been reduced from 89.66 s (Lexicographically based method) to 65.68 s (two layer block matching).
Image-II (Window): Image II in Fig. 6 with 430×470 pixels shows a nature photo including windows behind trees. The number of clusters created by the proposed method is 42. It took around 357.76 s to detect the forgery, while the lexicographically based method in the same situation need 490.82 s. Here, the proposed local block-matching algorithm leads to a 27.1% time improvement.
Image-III (Sky): Image III in Fig. 6 with 250×280 pixels is a photo of sky and clouds. Because of the low image complexity, the proposed method created only four clusters. The forgery detection time with the proposed method is around 136.27 s, while the lexicographically based method in the same situation need 152.99 s. In this case, the rate of improvement is 10.9%.
Image-IV (Tiled photo): Fig. 6 (Image IV) shows an image with 300×300 pixels includes several tiled photos. This image is a complex image and the best demonstration of performance of our algorithm. The number of clusters is 979. Hence, the computational time in this case is much better than lexicographically based method. Here, the times of detecting forgery are 94.60 s and 202.07 s for the proposed and lexicographically based method respectively. The rate of time complexity improvement is about 53.1%
Image-V (GolfGrass): Fig. 6 (Image V) with 1000×800 pixels shows a golf-grass image. When the block size was 24×24, the proposed method could not create more than 6 clusters because of the large block size and nature of the photo (golf grass). This is the worst case of our algorithm as the clustering algorithm is not able to group the blocks effectively. In this case, proposed two layer block matching performs several extra operations such as low accurate feature extraction and clustering. Hence, the computational time of our method is increased. For this photo, the running time of the proposed local block matching is 3399.9 s, while the lexicographically based method in the same situation need 2869.8 s.
Image-VI (Tree): The street stand-light in Fig. 7 (Tree) is the forged area. For detecting these duplicated regions the proposed coarse-to-fine method take around seconds, while lexicographically sorting needs seconds. It means, for this case the proposed method is superior to the common method.
Image-VII (Books): In the Bookstore image, because of the nature of the photo, there are several discriminative parts. Hence, the proposed method can do the clustering well. The two-layer block-matching algorithm here created 1102 clusters. Hence, we expect the high amount of improvement in computational time for this image.
Image-VIII (Home): For the image VIII (duplicated doors), the block size is supposed to be 16×16. This is a reason for reducing the ability of clustering with proposed algorithm in this case. If the block size is bigger, the clustering will be difficult and the algorithm cannot see the small textures.
Image-IX (TiledTexture): In the image IX (tiled textures photo) also we expect to have more clusters. Because the complexity of this image is high. The clustering algorithm could create 943 clusters with 46.6% improvement in computational time.
Image-X (Pigeons): In this image, the duplicated Pigeons should be detected. Although, the number of created cluster for this case is about 621, the time complexity of lexicographically sorting method excels to proposed method. Because the block size is 32×32. Hence, it can be a

Table 2
Performance of the proposed CTF block matching and lexicographical sorting for different images and different block size where dimension of low and high accurate feature vectors $\rho_1 = 1$, $\rho_2 = 64$.

Forged images			Results and evaluations			
Image	Image size	Block size	Lexico	Proposed CTF		Improved
			Time	ψ	Time	
I	227×191	8×8	89.66 s	26	65.68 s	↑26.7%
II	430×470	8×8	490.82 s	42	357.7 s	↑27.1%
III	250×280	8×8	152.99 s	4	136.27 s	↑10.9%
IV	300×300	8×8	202.07 s	979	94.60 s	↑53.1%
V	1000×800	24×24	2869.8 s	6	3399.9 s	↓−18.4%
VI	800×600	8×8	1254.6 s	59	906.02 s	↑27.7%
VII	800×600	8×8	1254.6 s	1102	655.7 s	↑47.7%
VIII	800×600	16×16	1399.5 s	95	1195.7 s	↑14.5%
IX	800×600	8×8	1254.6 s	943	669.1 s	↑46.6%
X	800×600	32×32	2001.7 s	621	2330.2 s	↓−16.4%
XI	800×600	8×8	1254.6 s	25	979.4 s	↑21.9%

fact that the bigger block size has negative affects in computational time of proposed algorithm.

Image-XI (Snow): For the image XI (duplicated a part of snow in the mountain), we change the block size to 8×8. The time complexity of proposed method is better than lexicographically sorting method about 21.9%.

Table 2 summarize the performance of the proposed Coarse-To-Fine (CTF) block matching algorithm compared with Lexicographical sorting (Fridrich, 2003).

5. Conclusion and future works

One of the major problems in copy–paste image forgery detection systems is the required high computational time of block matching step to find similar blocks. In this paper, we proposed a duplicated region detection model using block clustering with two-layer block matching algorithm. This algorithm needs two types of block feature: low accurate feature and high accurate feature. Low accurate feature is used for clustering the blocks in first match while high accurate feature is applied in local block matching (Second match). Because we just concentrate on improving the time complexity, the two feature extraction method are envisage as two black-box with feature vector dimension ρ_1 and ρ_2. Block clustering reduces the search space for exact block matching in second match which significantly improves the time complexity. The mathematical and experimental results demonstrate that when the number of clusters is greater than threshold $2^{\frac{b^{\rho_2}-\rho_2}{\rho_2-\rho_1}}$ time complexity of the local block matching is more efficient than lexicographically sorting algorithm. This case occurs when using small block size with high complex images. Moreover, a higher number of clustering leads to more improvement in our algorithm. Therefore, choosing a suitable Low accurate feature with lower feature vector dimension can increase the number of clusters and enhance the performance of the proposed algorithm. In future, we are going to extend our work to multilayer block matching with efficient multilayer feature extraction algorithms.

Acknowledgment

This research is supported by Universiti Teknologi Malaysia (UTM) using Fundamental Research Grant Scheme (FRGS) with vote number 78632.

References

Abramowitz, Stegun. Handbook of mathematical functions. McGraw-Hill; 1972.

Akbarpour Sekeh M, Maarof MAB, Rohani MF, Motiei M. Sequential straightforward clustering for local image block matching. World Academy of Science, Engineering and Technology 2011;74:775–9.

Bayram S. An efficient and robust method for detecting copy-move forgery. In: IEEE international conference on acoustics, speech and signal processing 2009.

Blanco JL, Gonzalez J, FernÂ'andez-Madrigal JA. An experimental comparison of image feature detectors and descriptors applied to grid map matching. Technical Report in Department of System Engineering and Automation 2010.

Bravo-Solorio S, Nandi AK. Automated detection and localisation of duplicated regions affected by reflection, rotation and scaling in image forensics. Signal Processing 2011;91(8):1759–70.

Christlein V, Riess C, Jordan J, Angelopoulou E. An evaluation of popular copy-move forgery detection approaches. Transactions on Information Forensics and Security 2012;7(6):1841–54.

Elena Deza MMD. Encyclopedia of distances – book. Springer; 2009.

Fridrich J. Detection of copy-move forgery in digital images. In: Proceedings of digital forensic research workshop 2003.

Huang Y, Lu W, Sun W, Long D. Improved DCT-based detection of copy-move forgery in images. Forensic Science International 2011;206: 178–84.

Jing L, Shao C. Image copy-move forgery detecting based on local invariant feature. Journal of Multimedia 2012;7(1):90–7.

Ledwich L, Williams S. Reduced sift features for image retrieval and indoor localisation 2004.

Li S, Wang Y. Multifocus image fusion using spatial features and support vector machine. Lecture Notes in Computer Science. Advances in Neural Networks 2005;3497:753–8. Springer Berlin/Heidelberg.

Lin HJ, Wang CW, Kao YT. Fast copy-move forgery detection. WSEAS Transactions on Signal Processing 2009;5:188–97.

Luo W. Robust detection of region-duplication forgery in digital image. Washington, DC, USA: IEEE Computer Society; 2006.

Mahdian B. A bibliography on blind methods for identifying image forgery. Signal Processing: Image Communication 2010;25:389–99. Elsevier.

Mahdian B, Saic S. "Detection of copy-move forgery using a method based on blur moment invariants.". Forensic Science International 2007; 171:180–9.

Michael Zimba SX. DWT-PCA (EVD) based copy-move image forgery detection. International Journal of Digital Content Technology and Its Applications 2011;5:251–8.

Nixon M, Aguado AS. Feature extraction and image processing. 2nd ed. Elsevier; 2008.

Pan X, Siwei L. Region duplication detection using image feature matching. Information Forensics and Security, IEEE Transactions 2010;5:857–67.

Pitas I. Digital image processing algorithms and applications. New York, NY, USA: John Wiley & Sons, Inc; 2000. ©2000.

Popescu, Farid H. Exposing digital forgeries by detecting duplicated image regions, TR2004-515. Dartmouth College, Computer Science; 2004.

Richard Neapolitan KN. Foundations of algorithms. 4th ed. Jones \& Bartlett Publishers; 2011.

Senjian A, Peursum P, Wanquan L, Venkatesh S. Efficient algorithms for subwindow search in object detection and localization. In: Computer vision and pattern recognition, 2009. CVPR 2009. IEEE Conference on 2009.

Sergois T, Koutroumbas. Pattern recognition – book. 4th ed. San diego: Elsevier; 2009.

Wiedermann J. The complexity of lexicographic sorting and searching. Applications of Mathematics 1981;26:432–6.

William Ford WRT. Data structures with Java. Pearson Education Inc; 2005.

Xiaoyuan X, Guoqiang H, Huaqing M. A novel algorithm for associative classification of image blocks. In: Computer and information technology, 2004. CIT '04. The fourth international conference on 2004.

Yang Q-C, Huang C-L. Copy-move forgery detection in digital image 2009.

Zhang C, Guo X, Cao X. Duplication localization and segmentation. In: Qiu G, Lam K, Kiya H, et al., editors. Lecture Notes in Computer Science, vol. 6297. Berlin – Heidelberg: Springer; 2010. p. 578–89.

Zhang Z. A survey on passive-blind image forgery by doctor method detection. In: Proceedings of the seventh international conference on machine learning and cybernetics. Kunming: IEEE; 2008.