Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/090253

Passive Forgery Detection for JPEG Compressed Image based on Block Size Estimation and Consistency Analysis

Cheng-Shian Lin[*] *and Jyh-Jong Tsay*

Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 621, Taiwan

Received: 24 Jun. 2014, Revised: 22 Sep. 2014, Accepted: 23 Sep. 2014
Published online: 1 Mar. 2015

Abstract: As most of digital cameras and image capture devices do not have modules for embedding watermark or signature, passive forgery detection which aims to detect the traces of tamping without embedded information has become the major focus of recent research for JPEG compressed image. However, our investigation shows that current approaches for detection and localization of tampered areas are very sensitive to image contents, and suffer from high false detection rates for localization of tampered areas for images with intensive edges and textures. In this paper, we present an effective approach which overcomes above problem, using reliable estimation and analysis of block sizes from the block artifacts resulting in JPEG compression process. We first propose an enhanced cross difference filter to strengthen block artifacts and reduce interference from edges and textures, and then integrate techniques from random sampling, voting and maximum likelihood method to improve the accuracy of block size estimation. We develop two different random sampling strategies for block size estimation: one for estimation of the primary JPEG block size, and the other for consistency analysis of local block sizes. Local blocks whose JPEG block sizes are different from the primary block size are classified as tampered blocks. We finally perform a refinement process to eliminate false detections and fill in undetected tampered blocks. Experiment over various tampering methods such as copy-and-paste, image completion and composite tampering, shows that our approach can effectively detect and localize tampered areas, and is not sensitive to image contents such as edges and textures.

Keywords: Passive image forgery detection, JPEG compression, JPEG block artifact extraction, maximum likelihood estimation

1 Introduction

Visual imagery has been widely used to provide essential evidences in many diverse areas, ranging from mainstream media, journalism and scientific publication, to medical imaging, criminal investigations and surveillance systems, to name a few. While we have historically had confidence on the integrity and authenticity of visual imagery, such trust has been gradually lost. With the rapid growth of digital devices and image editing technologies [1], [3], [16], [21], [22], [33] and [35], it has become easier than ever to produce and manipulate digital images with increasing sophistication. Doctored photographs are very difficult, if not impossible, to identify by visual examination. Digital image forensics which aims to verify the integrity and authenticity of digital images has thus become an important and exciting field of recent research.

There are two types of digital image forensics: active and passive. In active approaches, a watermark or signature which provides information to verify the integrity and authenticity of digital images is inserted into an image while it is acquired [13] and [31]. Unfortunately, many of the image capture devices do not contain the module to insert watermarks and signatures. Therefore, passive approaches which aim to detect traces of tampering without using prior information are extensively studied in recent research [9] and [32].

Over the past few years, a number of passive approaches for image forgery detection have been proposed, and can be roughly divided into five categories [9], namely, pixel-based, format-based, camera-based, physics-based and geometric-based. Pixel-based approaches examine pixel level anomalies caused by specific tampering, such as correlations between pixels arising from copy-and-paste [10] and [36], re-sampling

[*] Corresponding author e-mail: lchh95p@cs.ccu.edu.tw

[37], and splicing [4] and [34]. Format-based approaches exploit unique properties of image compression, such as block artifacts [12], [25], [26], [27], [28], [36], [38] in JPEG images. Camera-based approaches analyze the specific sensor artifacts caused by components in the imaging pipeline, such as color filter array interpolation (CFA) [2] and [39], camera response function [29], and sensor noise [7] and [8]. Physics-based approaches use physical rules to detect anomalies, such as lighting direction [18] and illumination constraint [30]. Geometric-based approaches inspect geometric properties of objects in the world and their positions relative to the camera [14], [15], [19], [44].

Notice that passive approaches based on pixels, cameras, physic and geometry are only applicable to uncompressed high quality images, and cannot effectively detect and locate tampering areas for JPEG images with high compression rates. Recently, several format-based approaches [1], [5], [6], [12], [23], [24], [25], [26], [27], [28], [36], [38], [43] for JPEG compressed images have been proposed. It has been noticed that block artifacts, resulting from JPEG compression process and appearing at JPEG block boundaries, are very useful for detection of tampering for JPEG images. Properties such as symmetry, periodicity and consistency derived from block artifacts are often destroyed when JPEG images are tampered. Previous research has studied the problem of double compression detection, quantization table estimation and localization of tampering areas. A JPEG image compressed twice is considered as being decompressed, tampered and then compressed again. Chen and Hsu [5],[6] use periodic properties of JPEG block artifact noise to detect image cropping and recompression. Luo et al. [23] exploited the symmetry of the blocking artifact characteristics matrix (BACM) to detect whether an image has been cropped and double compressed. Barni et al. [1] further integrated the characteristics of BACM and an image segmentation algorithm to localize tampered areas. However, the accuracy of tampering detection is sensitive to image content, and depends on algorithms for image segmentation which is an ill-posed problem.

Ye et al. [43] use the histogram power spectrum of DCT coefficients to estimate the original JPEG quantization table which is then used to identify tampered areas containing inconsistent block artifacts. However, their approach needs user to select correct regions for quantization table estimation, and assumes that the JPEG block size is 88. It should be noted that the JPEG format allows for block sizes other than 8×8, and the verifier may not know the block size of the image encoder in real circumstance [24].

Recently, Li et al. [25] presented a different approach which is based on extraction of block artifact grids (BAG). Their experiment demonstrated that their approach can successfully detect and localize tampered areas for several tampering methods such as image cropping, copy-and-paste and inpainting for JEPG images with block size 8×8. However, our experiment shows that their approach is sensitive to image content, and can suffer from high false detection rates for localization of tampered areas for images with intensive edges and textures.

In this paper, we present an effective and robust approach which is based on reliable estimation and analysis of JPEG block sizes, and can handle images with arbitrary block size. We first propose an enhanced cross difference filter to strengthen block artifacts and reduce interference from edges and textures, and then integrate techniques from random sampling and voting to improve the accuracy of the maximum likelihood method [24] for blind block size estimation. We develop two different random sampling strategies for block size estimation: one for estimation of the primary block size of the whole image, and the other for the local block sizes of small regions of the image. The purpose of local block size estimation is to verify artifact consistency between local regions and the whole image. Local regions fails to pass consistency verification are identified as tampered regions. We finally perform a refinement process which re-estimates local block sizes for small connected tampered regions to eliminate false detections, and for small connected un-tampered regions to fill in undetected tampered blocks.

We have carried out experiment over major tampering methods such as copy-and-paste, image completion and composite tampering, and the result shows that our approach outperforms previous approaches [24] and [25], and can effectively detect and localize tampered areas even for images with intensive edges and textures.

The rest of this paper is organized as follows. Section 2 briefly overviews the problem, and sketches our main approach. Section 3 presents details for primary block size estimation. Section 4 presents details for tampered block detection and detection result refinement. Section 5 presents the experimental results. Section 6 concludes.

2 The problem and approach overview

In this section, we briefly overview the problem of tampering detection in JPEG compressed images, and sketch our proposed approach.

JPEG compression is the most widely used compression method for still image. In the JPEG encoding procedure, an image is first partitioned into 8×8 non-overlapping blocks. The Discrete Cosine Transform (DCT) is then applied to each block, and the DCT coefficients are quantized. Finally, the quantized DCT coefficients are entropy encoded and output as part of the compressed image data [40]. It should be noted that the JPEG format allows for DCT block sizes other than 8×8. For example, version 8 of the JPEG software provides arbitrary block sizes from 1×1 to 16×16 pixels [45]. In general, large block sizes for smooth images will get higher compression. On the contrary, small block sizes

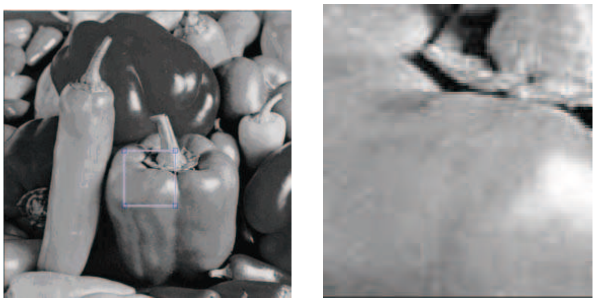
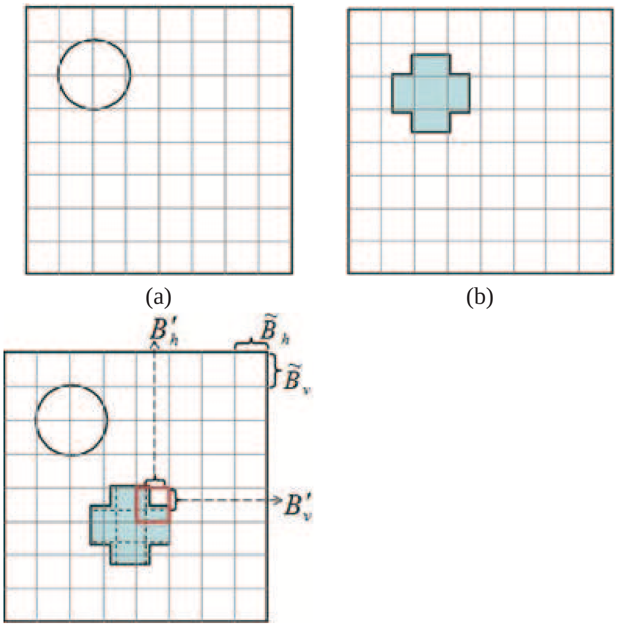


Fig. 1: Close view of block artifacts for JPEG compressed image "Peppers" with QF=50.

for high-detail images will get higher image quality but lower compression.

Since the quantization process of JPEG compression is performed on each block of image independently, blocking artifacts (noises), as shown in Fig. 1, will appear at block boundaries between adjacent blocks in the image. Although the artifact degrades the quality of the image, it has been widely applied as an intrinsic feature for authenticity and integrity verification in forensic analysis. Extraction and analysis of block artifacts have played an essential role in detection and location of tampered areas for JPEG image. Fig. 2 gives an illustration of copy-and-paste tampering which is performed to copy a region from a source image and pastes it to a target image to form a tampered image. It has been noticed [1], [23], [25] and [26] that in order to create a plausible tampered image to illude human eyes, the source image must be pasted in proper place in the target image, and hence the JPEG block boundaries in the tampered region, represented as dashed lines in Fig. 2(c), are usually mismatched with those in un-tampered regions. Block boundary inconsistency appearing in tampered regions is a crucial evidence to detect and localize tampered regions.

In this paper, we aim to develop an effective approach to detection and localization of tampered for JPEG images manipulated by copy-and-paste, image completion and composite tampering. We propose an approach based on reliable estimation and analysis of JPEG block sizes. As shown in Fig. 3, the proposed approach consists of three major steps: 1) primary block size estimation, 2) local block size estimation and tampered area detection, and 3) detection result refinement. We first propose an enhanced cross difference filter to produce a block boundary noise map ($BBNM$) which strengthens block artifacts and reduces strong edges and textures. Period signals are then extracted from the noise map for estimation of the primary block size $\widetilde{B} = (\widetilde{B_v}, \widetilde{B_h})$. We integrate techniques from random sampling, voting and maximum likelihood methods to obtain reliable estimation of JPEG block size. We then partition the noise map into sub-maps of size $3\widetilde{B_v} \times 3\widetilde{B_h}$ each, and for each sub-map, estimate its local block size $B' = (B'_v, B'_h)$, and classified it as tampered if $B' \neq \widetilde{B}$, i.e. $B'_v \neq \widetilde{B_v}$ or $B'_h \neq \widetilde{B_h}$ as shown in Fig. 2(c). We finally perform a refinement process to eliminate false detections and fill-in undetected tampered blocks. Note that, for computational efficiency, we convert the color image to grayscale, using the luminosity method which is adopted in [20]. Based on human perception, the method gives high weight to green component than other color components, and computes the grayscale value of each color pixel as follows:



(a)

(b)

(c)

Fig. 2: (a) the target image; (b) the source JPEG image; (c) a tampered image with the blue region pasted from source JPEG image.

$$\text{gray value} = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B, \quad (1)$$

where R, G, and B denote the R, G, and B components of the pixel, respectively. Details of each step of our proposed approach will be given in next subsequent sections.

3 Primary block size estimation

In this section, we present details of our approach for estimation of JPEG block sizes for JPEG compressed images.

A maximum likelihood estimation (MLE) approach for blind estimation of JPEG block size was proposed by Lin et al. in [24]. The basic idea in MLE is to first define an intensity difference filter to capture block artifact boundaries, then aggregate the difference along each dimension to obtain 1-D signals, and finally estimate the

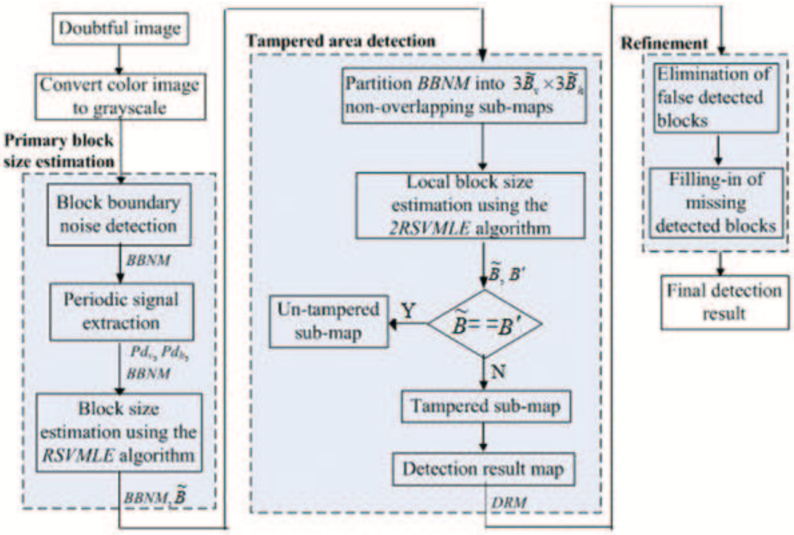

Doubtful image
Convert color image to grayscale
Primary block size estimation
Block boundary noise detection
BBNM
Periodic signal extraction
Pd_x, Pd_y, BBNM
Block size estimation using the RSVMLE algorithm
BBNM, B̃
Tampered area detection
Partition BBNM into 3B̃_x ×3B̃_y non-overlapping sub-maps
Local block size estimation using the 2RSVMLE algorithm
B̃, B'
B̃==B'
Y
N
Un-tampered sub-map
Tampered sub-map
Detection result map
DRM
Refinement
Elimination of false detected blocks
Filling-in of missing detected blocks
Final detection result

Fig. 3: The proposed passive forgery detection approach.

period of the 1-D signals, which corresponds to the block size, using maximum likelihood estimation. However, our experiment shows that *MLE* is highly sensitive to image content, and suffers from poor accuracy for images with intensive edges and textures. In this section, we present a new approach *RSVMLE* which substantially improves the estimation accuracy of *MLE* by integrating techniques from random sampling, voting and maximum likelihood estimation.

3.1 Edge interference reduction

Let I be an $M \times N$ input image, and $I(x,y)$ be the intensity value of pixel (x,y), where $x \in \{1,...,M\}$ and $y \in \{1,...,N\}$. The *MLE* method in [24] uses the difference filter $d(x, y)$ defined as $d(x,y) = I(x,y) - I(x-1,y)$ for pixel (x,y). However, the difference filter is highly interfered by edges and textures in the image. Since most edges are neither vertical nor horizontal, a cross difference filter $g(x,y)$ is proposed in [12] to reduce interference from edges, and is defined as follows.

$$g(x,y) = |I(x,y) + I(x+1,y+1) - I(x+1,y) - I(x,y+1)|, \tag{2}$$

However, as shown in Fig. 4(b) and 4(c), although the cross difference filter improves the difference filter, in both of them, a large portion of block artifact boundaries is still very weak due to strong edges in the image. To further weaken strong edges and strengthen block artifact boundaries, we propose the following enhanced cross difference filter $f(x,y)$.

$$f(x,y) = \begin{cases} \alpha \cdot g(x,y), & \text{if } \mathrm{g}(x,y) > \theta, \\ (1-\alpha) \cdot g(x,y), & \text{otherwise.} \end{cases} \tag{3}$$

where θ is a threshold value to classify noises from strong edges, and α is a reducing coefficient to reduce noises from strong edges. In this paper, θ and α are empirically set to 15 and 0.2, respectively. Note that our observation shows that the difference value of a pixel at block artifact boundary is rarely above 15. Setting θ=15 and α=0.2 is expected to weaken strong edges while strengthening block artifact boundaries so that block artifact boundaries can be reliably captured. Fig. 4(c) shows the result of cross difference filter [12] in which the contrast has been enhanced for clarity. Although the cross difference filter can detect block artifact boundaries, it still introduces a large amount of noise. Fig. 4(d) shows that the enhanced cross difference filter greatly reduces the influence of noise, and can be used to obtain a reliable estimation of the block size. It should be noted that we have carried out experiments over different values of θ, ranging from 0 to 55, and θ=15 achieves the best detection and localization performance as discussed in section 5.3.

3.2 Reliable maximum likelihood block size estimation

The block size is estimated by maximum likelihood estimation on the block boundary noise map (*BBNM*) with *BBNM*(x,y)=$f(x,y)$ produced by the enhanced cross difference filter. To obtain reliable estimation, we apply random sampling to create many instances of slightly different noise maps, then run maximum likelihood estimation for each sampled map, and finally perform voting to decide the most reliable block size.

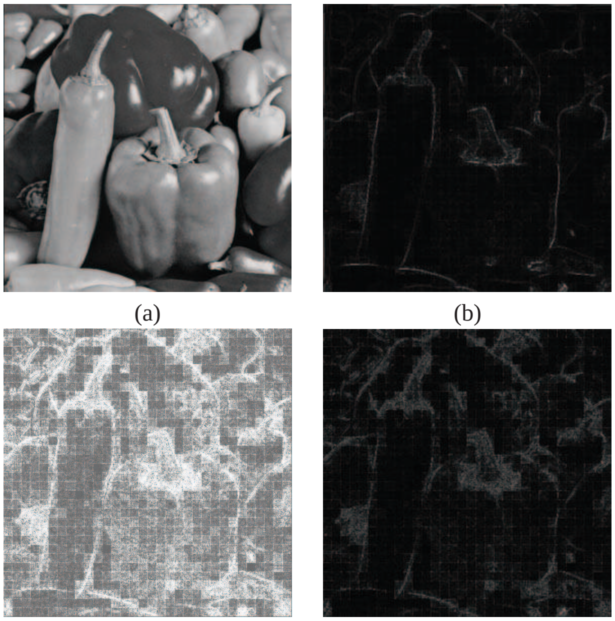

(a) (b) (c) (d)

Fig. 4: JPEG compressed "Peppers" image with a block size of 16, (b) the result of difference filter [24], (c) the result of cross difference filter [12] in which contrast has been enhanced, (d) the result of the proposed enhanced cross difference filter.

3.2.1 Periodic signal extraction

Our first step of block size estimation is to extract 1-D periodic signals from each sampled noise map.

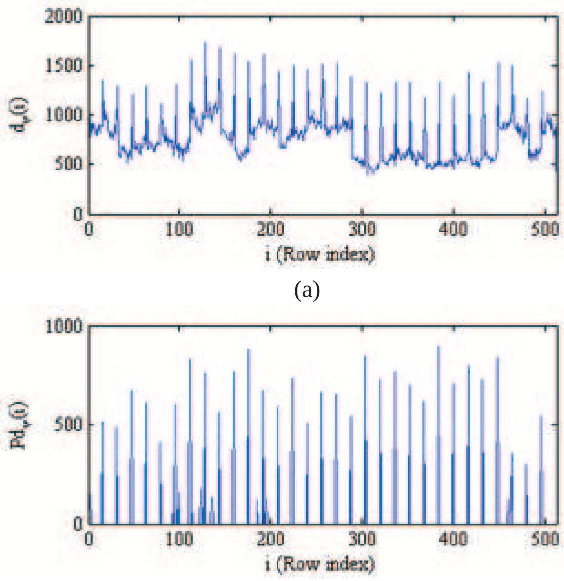
Let $BBNM^*$ be a randomly sampled noise map. $BBNM^*$ is generated by a random number generator as follows.

$$BBNM^*(x,y) = \begin{cases} BBNM(x,y), & \text{if } random() > \tau_1, \\ 0, & \text{otherwise}, \end{cases} \qquad (4)$$

where $random()$ is a random number generator which uniformly generates a random number between 0 and 1. Note that τ_1 is a threshold value calculated by τ_1=1/log$_2$($\sqrt{M \times N}$). In this study, we simply set τ_1 to be 0.1 which is close to the values computed for sizes of all the images in our experiment. The setting achieves 90% expected sampling rate, and keeps each sampled noise map highly similar to the original map, so that the block size can be reliably estimated.

For each sampled noise map $BBNM^*$, we compute two 1-D signals, one vertical d_v and one horizontal d_h, by summing the difference values in $BBNM^*$ along the vertical and horizontal directions, respectively. Signal d_v is defined as follows. Signal d_h is defined similarly.

$$d_v(i) = \sum_{x=1}^{M} BBNM^*(x,y), \quad i = \{1,\ 2, \ldots, M\}. \qquad (5)$$



(a)

(b)

Fig. 5: Signals d_v and Pd_v for JPEG compressed "Peppers" image with a block size of 16: (a) Signal d_v with peaks at multiple of 16, (b) Signal Pd_v with peaks strengthened and noises reduced.

Note that as in Fig. 5(a), when the block size in horizontal direction is B_h, d_v will have approximately periodic signals with peaks at multiples of B_h. We next explain how to estimate the period of d_v. The period of d_h can be estimated in a similar fashion.

To further reduce noise influence for period estimation, we compute Pd_v based on the first derivative of $d_v(i)$, i.e. $d'_v(i)$=$\partial d_v(i)/\,\partial i$, as follows.

$$Pd_v(i) = \begin{cases} d'_v(i), & \text{if } d'_v(i) > 0 \text{ and } d'_v(i) > \lambda_v, \\ 0, & \text{otherwise}, \end{cases} \qquad (6)$$

where λ_v is a threshold value calculated by averaging positive signals in d'_v. Fig. 5 shows the signals d_v and Pd_v for the 512×512 image "Peppers" in Fig. 4(a), which has been JPEG compressed with a block size of 16. The magnitude of signal d_v is the strength of block artifact boundary noise. Signal Pd_v has the same period with d_v, and enhances d_v with strengthened peaks and reduced noises. Similarly, we compute Pd_h from d_h.

Figure 5 shows the signals d_v and Pd_v for the 512×512 test image "Peppers" in Fig. 4(a), which has been JPEG compressed with a block size of 16. Obviously, as shown in Fig. 5(a), the magnitude of signal d_v is the strength of block artifact boundaries noise. By calculating of the first derivative and Eq. (6), the signal Pd_v is approximately periodic, as shown in Fig. 5(b). More specifically, we expect the block size of image can be determined by the period of the signal Pd_v.

3.2.2 MLE signal period estimation

We apply the maximum-likelihood estimation (MLE) scheme [24] and [42] to estimate the periods of signal Pd_v and Pd_h, which correspond to the block sizes in horizontal and vertical directions, respectively. Suppose that the signal Pd_v comprises periodic signal s plus an $i.i.d$ Gaussian noise n with a mean of zero, i.e.

$$Pd_v(i) = s(i) + n(i), \quad i \in \{1, 2, \ldots, M\}, \tag{7}$$

where s is a periodic repetition of a signal q with the period B_h. Namely,

$$s(i) = q(i \; mod \; B_h). \tag{8}$$

To estimate period B_h from the periodic signal Pd_v, the maximum likelihood estimation [24] and [42] maximizes the conditional probability density function $P(Pd_v|s, \sigma_2, B_h)$, with respect to signal parameter s, noise variance σ_2, and period B_h, by minimizing the estimated noise variance $\hat{\sigma}^2(B_h)$ as a function of B_h. Let B_h^{MLE} be the MLE estimation of B_h.

$$B_h^{MLE} = \arg\min_{B_h} \hat{\sigma}^2(B_h). \tag{9}$$

However, MLE estimation still can be affected by noises in the map $BBNM$ caused by edges and textures. We apply voting from randomly sampled noise maps to improve the reliability of MLE estimation. Let H_{vote} be the period voting histogram, and $\widetilde{B_h}$ be the final estimation of the block size.

$$\widetilde{B_h} = \arg\max_{B_h^{VML}} \{ \sum_{k \in iter} H_{vote}(B_h^{VML}) \}. \tag{10}$$

Algorithm 1 summarizes the main steps of the proposed algorithm, $RSVMLE$, which integrates random sampling, voting and maximum likelihood estimation for reliable block size estimation. In our experiment, the number of iteration is 30.

Algorithm 1 (The proposed $RSVMLE$ algorithm)

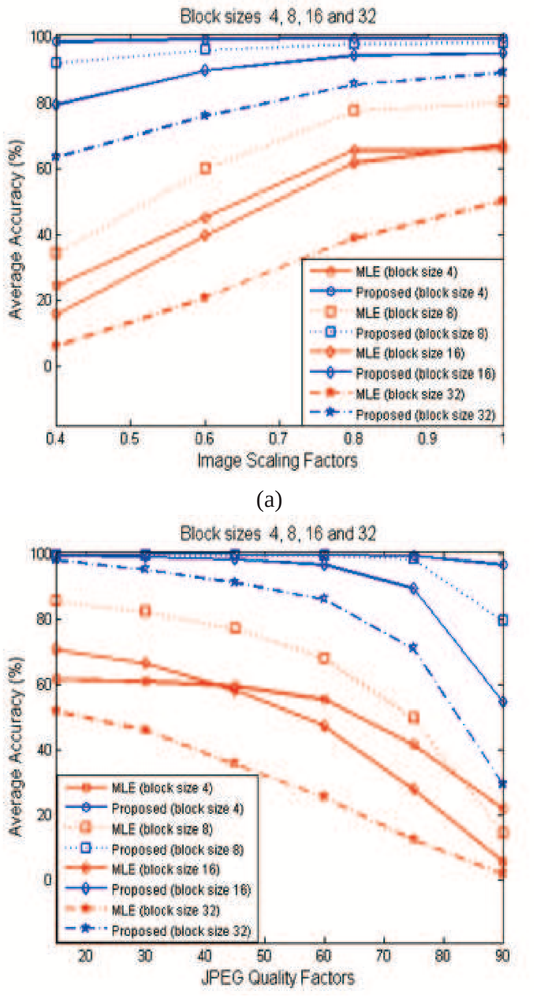
```
Input:
    the block boundary noise map $BBNM$.
Output:
    the block size $\widetilde{B_h}$ in horizontal direction.
1. Randomly sample $BBNM^*$ from $BBNM$ as follows.
   $BBNM^*(x,y) = \begin{cases} BBNM(x,y), & \text{if } random() > \tau_1, \\ 0, & \text{otherwise,} \end{cases}$
2. Compute signal $d_v$ and $Pd_v$ from $BBNM^*$ as follows.
   $d_v(i) = \sum_{x=1}^{M} BBNM^*(x,y), \quad i = \{1, 2, \ldots, M\}.$
   $Pd_v(i) = \begin{cases} d_v'(i), & \text{if } d_v'(i) > 0 \text{ and } d_v'(i) > \lambda_v, \\ 0, & \text{otherwise,} \end{cases}$
3. Estimate the period $B_h^{MLE}$ using the maximum-likelihood estimation.
   $B_h^{MLE} = \arg\min_{B_h} \hat{\sigma}^2(B_h).$
4. Vote for the period $B_h^{MLE}$ in the period histogram.
5. Repeat steps 1 to 4 until the predefined number of iterations is reached.
6. Output $\widetilde{B_h}$ which is the period with maximum number of voters.
   $\widetilde{B_h} = \arg\max_{B_h^{VML}} \{\sum_{k \in iter} H_{vote}(B_h^{VML})\}.$
```

Note that the block size $\widetilde{B_v}$ in vertical direction can be estimated similarly by the proposed $RSVMLE$ algorithm.

3.2.3 Performance of RSVMLE

To evaluate the performance of algorithm $RSVMLE$, we have carried out experiments over 885 images, each of size 512×384, from an uncompressed color image database UCID [41]. This investigation measures the estimation accuracy for combinations of different image scaling factors, JPEG block sizes and quality factors. The scaling factor scales image size. Small parameter values indicate that original image is downscaled to smaller levels. We use the built-in image resizing function in Matlab software to create four groups of different sizes with scaling factors 0.4, 0.6, 0.8, and the original size, respectively. In JPEG compression, we examine four block different sizes, including 4×4, 8×8, 16×16, and 32×32, and six quality factors, including 15, 30, 45, 60, 75, and 90.

We compare our approach $RSVMLE$ with the original MLE proposed in [24]. Fig. 6 (a) gives the average accuracy for different image scaling factors, and Fig. 6 (b) gives the average accuracy for different JPEG quality factors. Both figures show that $RSVMLE$ outperforms MLE. $RSVMLE$ achieves average accuracy 91.09% which is nearly twice of the accuracy achieved by MLE. Fig. 6 (a) also shows that the accuracy increases as the number of blocks creases, i.e. as the image size increases or the block size decreases. Fig. 6 (b) shows that the accuracy decreases as the quality factor increases. This is



(a)

(b)

Fig. 6: (a) Average accuracy for different image scaling factors; (b) average accuracy for different JPEG quality factors.

because the artifact is weakened as the image quality increases.

4 Tampered block detection and refinement

The process of tampered block determination consists of two processes: tampered block detection and detection result refinement. In tampered block detection, we perform local block size estimation for each block to identify blocks whose sizes are different from the primary block size $(\widetilde{B_v},\widetilde{B_h})$ estimated in previous section. Those blocks are classified as tampered. The refinement process aims to improve the detection result by eliminating false detections and filling in undetected tampered blocks.

4.1 Tampered block detection

We explain how to determine whether a block is tampered or not. Let $BBNM$ be the noise map of an $M \times N$ JPEG image, and $(\widetilde{B_v},\widetilde{B_h})$ be the primary block size estimated by $RSVMLE$. In the detection process, we first partition $BBNM$ into $\lfloor M/3\widetilde{B_v} \rfloor \times \lfloor M/3\widetilde{B_h} \rfloor$ non-overlapping sub-maps of size $3\widetilde{B_v} \times 3\widetilde{B_h}$ each, and assigns a 2-dimensional index (i,j) to each block, with $1 \le i \le \lfloor M/3\widetilde{B_v} \rfloor$ and $1 \le j \le \lfloor M/3\widetilde{B_h} \rfloor$. Let $bbnm_{i,j}$ denote the sub-map of $BBNM$ with index (i,j). Namely, sub-map $bbnm_{i,j}$ is composed of all $BBNM(x,y)$ for for $(i-1)3\tilde{B}_v + 1 \le x \le (i+1)3\tilde{B}_v$ and $(j-1)3\tilde{B}_h + 1 \le y \le (j+1)3\tilde{B}_h$, as shown in Fig. 7.

Our next step is to estimate the local block size (B'_v,B'_h) of each sub-map $bbnm_{i,j}$. (B'_v,B'_h) is estimated by $2RSVMLE$ given in Algorithm 2, which is similar to $RSVMLE$, but adopts a different sampling strategy to achieve reliable estimation from small noise maps as follows. Let $bbnm^*_{i,j}$ be a random sample of $bbnm_{i,j}$, for estimation of B'_v.

$$bbnm^*_{i,j}(x,y) = \begin{cases} bbnm_{i,j}(x,y), & \text{if } mod(x,\tilde{B}_v) == 0 \\ & and \;\; mod(y,\tilde{B}_h) == 0, \\ bbnm_{i,j}(x,y), & \text{if } mod(x,\tilde{B}_v) \neq 0 \\ & and \;\; mod(y,\tilde{B}_h) \neq 0 \\ & and \;\; \text{if} \;\; randndom() > \tau_2, \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

where 1/(log$_2(\widetilde{B_v})$-0.1) is a function of $\widetilde{B_v}$ Note that when primary block size B_v increases, threshold τ_2 will decrease, and hence the sampling rate will increase, and vice versa.

```
Algorithm 2 (The proposed $2RSVMLE$ algorithm)
Input:
    the examined sub-map $bbnm_{i,j}$, and
    the blocking artifact boundary positions $(\widetilde{B_v},\widetilde{B_h})$.
Output:
    the block size of the examined area $B'_h$.
1. Randomly sample $bbnm^*_{i,j}$ from sub-map
    $bbnm_{i,j}$ as follows.
    $bbnm^*_{i,j}(x,y) =$
    $\begin{cases} bbnm_{i,j}(x,y), & \text{if } mod(x,\tilde{B}_v) == 0 \\ & and \;\; mod(y,\tilde{B}_h) == 0, \\ bbnm_{i,j}(x,y), & \text{if } mod(x,\tilde{B}_v) \neq 0 \\ & and \;\; mod(y,\tilde{B}_h) \neq 0 \\ & and \;\; \text{if} \;\; randndom() > \tau_2, \\ 0, & \text{otherwise}, \end{cases}$
2. Compute the signal $d_v$ and $Pd_v$ from $bbnm^*_{i,j}$.
3. Estimate the period $B_h^{MLE}$ of using $Pd_v$ the
    maximum-likelihood estimation.
4. Vote for the period $B_h^{2VML}$ in the period histogram.
5. Repeat steps 1 to 4 until the predefined number of
    iterations is reached.
6. Acquire the block size of the examined area $B'_h$
    from the period bin containing the maximum value
    using Eqs.(10)-(11).
```

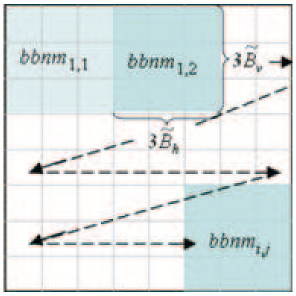
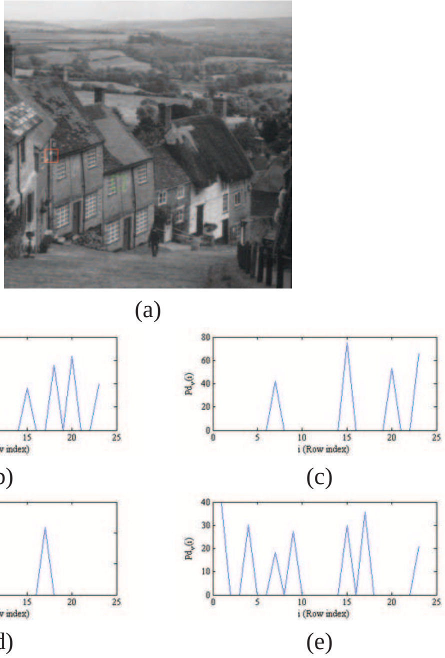


Fig. 7: Illustration of sub-maps.

All blocks in sub-map $bbnm_{i,j}$ are detected as tampered blocks if the block size of $bbnm_{i,j}$ is different from the primary block size, i.e. $(B'_v,B'_h)\neq(\widetilde{B_v},\widetilde{B_h})$ We develop another algorithm $2RSVMLE$ to estimate local block size (B'_v,B'_h), which is similar to $RSVMLE$ but uses a different sampling strategy to achieve reliable analysis for small noise maps, as shown in algorithm 2.

Fig. 8 gives an example to show how algorithm $2RSVMLE$ estimates the block sizes of tampered as well as un-tampered sub-maps. As shown in Fig. 8(a), the red block represents an un-tampered area, and the green block represents a tampered area. The primary block size is 24×24. Fig. 8(b) shows the original projection signal Pd_v of the un-tampered area, for which the estimated block size by MLE is 4 because of the periodicity property of signal Pd_v is destroyed by edge noise influence. On the contrary, as shown in Fig. 8(c), the different sampling strategies can reduce the influence of edge noise to maintain the periodic property of signal Pd_v. Thus, the accurate block size 8 can be obtained via the MLE scheme. Additionally, if the blocking artifact boundaries of the tampered region are mismatched with that of the blocking artifact boundaries of the un-tampered region, the different sampling strategies can also make the signal Pd_v non-periodic which will result in misjudgments, as shown in Fig. 8(d)-(e), the block size is calculated as 2. As can be seen, the proposed different sampling strategies can effectively help the proposed $2RSVMLE$ algorithm calculate the block size of areas with small size and high texture.

Algorithm 2 gives the main steps in $2RSVMLE$ for local block size estimation, and algorithm 3 gives the main steps for tampered block detection, which returns a detection result map (DRM) which will be refined to get the final detection result.



(a)

(b) (c)

(d) (e)

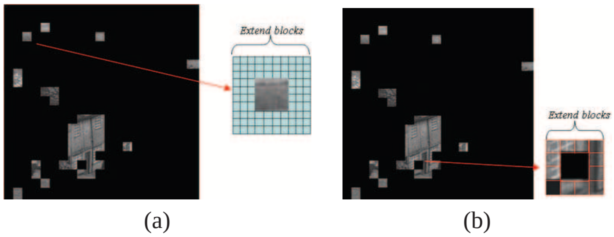
Fig. 8: The proposed difference sampling strategies for “small” examined area of forged image: (a) the investigated un-tampered and tampered areas, denoted by a red box and green box respectively, (b) signal Pd_v of un-tampered area computed by MLE, (c) signal Pd_v of the proposed approach for un-tampered area computed by $2RSVMLE$, (d) signal Pd_v of tampered area computed by MLE, (e) signal Pd_v of the proposed approach for tampered area computed by $2RSVMLE$.

Algorithm 3 (The proposed tampered block detection algorithm)

```
Input:
    the $BBNM$, and the primary block size $(\widetilde{B_v},\widetilde{B_h})$.
Output:
    the detection result map($DRM$)
1. Partition $BBNM$ into $\lfloor M/3\widetilde{B_v}\rfloor \times \lfloor M/3\widetilde{B_h}\rfloor$ non-overlapping sub-maps of size $3\widetilde{B_v} \times 3\widetilde{B_h}$ each, and examine each sub-map $bbnm_{i,j}$.
2. For each sub-map $bbnm_{i,j}$, repeat steps 3 and 4 for 7 times.
3. Estimate the local block size $(B'_v,B'_h)$ of sub-map $bbnm_{i,j}$ using algorithm $2RSVMLE$.
4. $DRM(i,j) = 1$ if the estimated local block size of $bbnm_{i,j}$ is different from the primary block size, i.e. $(B'_v,B'_h) \neq (\widetilde{B}_v,\widetilde{B}_h)$, and $DRM(i,j) = 0$, otherwise.
5. Output the detection result map $(DRM)$.
```

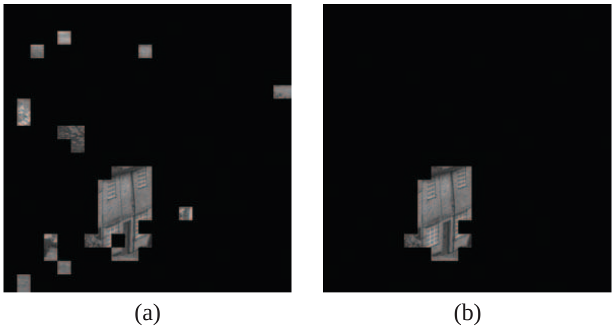
4.2 Detection result refinement

The main idea for result refinement is to re-verify block size consistency for all small connected tampered or un-tampered regions whose enclosing rectangles are smaller than 72×72 pixels.



(a) (b)

Fig. 9: Illustration of refinement: (a) elimination of false detection; (b) filling-in undetected tampered blocks.



(a) (b)

Fig. 10: (a) the detection result before refinement, (b) the result after refinement.

As in algorithm 4, to eliminate false detections, we first identify all regions of connected tampered blocks from the detection result map (*DRM*), using the 4-connected component labeling algorithm, which is developed to detect connected regions in binary images. Then, for each small connected region with enclosing rectangle smaller than 72×72, we expand the region with size 24 pixels in each direction of left, right, bottom and up as shown in Fig. 9(a), and re-estimate the local block size of the expanded region, using algorithm $2RSVMLE$. If the re-estimated block size is equivalent to the primary block size, we conclude that the detection is false, and eliminate the region from the final list of tampered blocks. Similarly, to fill in undetected tampered blocks, we identify all small un-tampered regions, and for each small region, count the number of tampered blocks in the expanded region. If the ratio of tampered blocks in the expanded region is larger than 0.7, the blocks in the small region is detected as tampered and added to the final detection list of tampered blocks. Figure 10 shows the results for the 512×512 test image “Gold hill” with and without refinement. As shown in Fig. 10(b), the refinement process does eliminate false detections, and fill in un-detected tampered blocks.

```
Algorithm 4 (The proposed refinement algorithm)
Input:
    $DRM$, $BBNM$, and ($\widetilde{B_v}$,$\widetilde{B_h}$).
Output:
    the final detection result map ($FDRM$)
(Steps 1 to 4 are for false detection elimination.)
1. Identify all connected tampered region by connected
    component labeling on the tampered graph defined
    from $DRM$ as follows. Each entry $(i,j)$ corresponds
    to a node in the graph if $DRM(i,j)=1$, and has
    edges connected to its 4 neighbors at left, right,
    bottom and up if their corresponding blocks are
    also marked as tampered.
2. For each connected region identified in Step 1,
    compute its smallest enclosing rectangle, and mark
    it “small” if the length or the width of the rectangle
    is smaller than 72.
3. For each “small” connected region identified in step 2,
    expand the region 24 pixels at its left, right, bottom
    and up. Namely, if its smallest enclosing rectangle is
    from rows $i_1$ to $i_2$ and columns $j_1$ to $j_2$, the
    expanded region is the rectangular region from rows
    $i_1-24$ to $i_2+24$ and columns $j_1-24$ to $j_2+24$.
4. For each expanded region, re-verify its consistency
    by re-estimating its local block size, using
    Algorithm $2RSVMLE$. If the re-estimated local
    block size is equivalent to the primary block size,
    we conclude that the detection is false, and eliminate
    the detection from the final result.
(Steps 5 to 7 are for filling in undetected tampered
    blocks.)
5. Identify all connected un-tampered region by
    connected component labeling on the tampered
    graph defined from the complement of $DRM$ as in
    Step 1.
6. For each connected un-tampered region identified
    in Step 5, compute its smallest enclosing rectangle,
    and mark it “small” if the length or the width of its
    enclosing rectangle is smaller than 72 pixels.
7. For each “small” connected un-tampered region
    identified in Step 6, expand the region as in Step 3,
    and count the ratio of tampered blocks in that region.
    If the ratio is larger than 0.7, we conclude that the
    region is tampered, and is added to the final list of
    tampered blocks.
8. Output the final detection result map ($FDRM$).
```

5 Experimental results

In this section, we first give several examples to illustrate the effectiveness of our approach. We then give the performance measured from 780 test images, and compare our result to the *MLE* which is proposed in [24] for block size estimation and extended by this study for tampered block detection, and *BAG* [25] which is based on extraction of block artifact grids. All experiments are run on a PC with an Intel Core i7-920 CPU 2.67GHz and 4G RAM, using the Matlab software development tool. Full details of all test images and experiments are available at our website [46].

5.1 Examples and Discussion

In this subsection, we gives the detailed results of 5 test images with various content types such as smooth regions, high texture, strong edges, and intensive edges. Table 1 lists the main characteristics of the 5 test images. "Gold Hill" is an image with intensive edges, "Nature Scene" is an image with high texture, "Beach" is an image with smooth regions and high texture areas, "Campanile " is a smooth photo, and "Battlefield" is an image with smooth regions, strong edges and high texture areas.

Note that, we carry out experiment over images manipulated by various image editing methods such as copy-and-paste, image completion and composite tampering. Copy-and-paste is a common method which copies an area from a source image and pastes it to a target image. Image completion [3] and [21] aims at filling in missing pixels in a large unknown region of an image, caused by the removal of large objects, in a visually plausible way. Composite tampering generally performs various types of image tampering processes, including image completion [3] and [21], copy-and-paste, and seamless cloning [35] to create a seamless and convincing fake image which often changes the content of the original image. To create tampered image, we use Photoshop for copy-and-paste, and implement the approach in [21] for image completion, and the approach in [35] for seamless cloning. In addition, we also simulate variable block sizes tampering (VBS) in which the block sizes of the source and target images are different. It should be noted that the JPEG format allows for block sizes other than 8×8, and the verifier may not know the block size of the image encoder in real circumstances [24].

1) Copy-and-paste tampering: Figure 11 gives the result of the experiment for the image "Gold Hill" which has intensive edges and JPEG block size 8×8. In the experiment, the source and target images are the same image with the same block size. Fig. 11(a) gives the original image in which the area surrounded by the read curve is to be tampered, and Fig. 11(b) gives the tampered image in which the grey wall area of the original image is pasted to the neighboring white wall area. Fig. 11(c), (d) and (e) show the detected areas by *MLE* [24], *BAG* [25], and our approach. The results show that our approach is the only approach which effectively detects the tampered area as shown in Fig. 11(e), and outperforms the other two approaches. Note that the extension of *MLE* has very high false detection rate as it cannot correctly estimate local block sizes, due to interference from intensive edges. *BAG* is also vulnerable to interference from intensive edges, and results in false detection around the tampered area. In the comparison, we compute average intensity, and classify a block as tampered if its intensity is above the average, and un-tampered, otherwise.

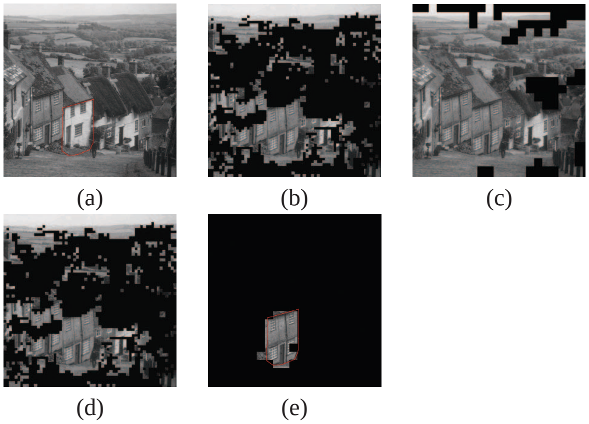


Fig. 11: The copy-and-paste forged image: (a) the original image, "gold hill;" (b) the tampered image; (c)-(e) the detection results by extension of *MLE* [24], *BAG* approach [25], and our approach, respectively.

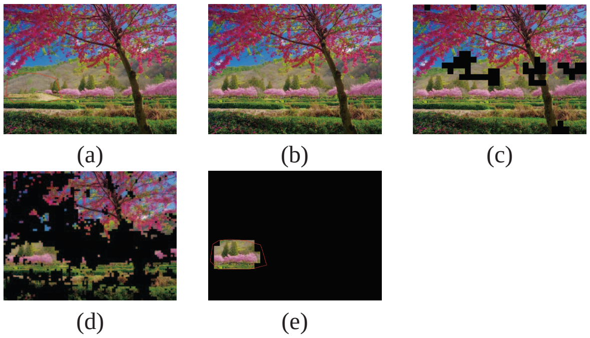


Fig. 12: The copy-and-paste forged image: (a) original image, "nature scene;" (b) the tampered image; (c)-(e) the detection results by *MLE* [24], *BAG* approach [25], and our approach, respectively.

Figure 12 shows the result for image "Nature Scene" whose content has high texture. The source and target images are the same image with the same block size 8×8. Fig. 12(a) gives the original image in which the area surrounded by the read curve is the region to be tampered, and Fig. 12(b) gives the tampered image in which the central area of thick grass and trees with high texture is pasted to the hillside area on its left. Notice that in Fig. 12(b), the tampered area is hardly identified by visual examination. Fig. 12(c), (d) and (e) show the areas detected by *MLE*, *BAG* and our approach. The results show that our approach outperforms the other two.

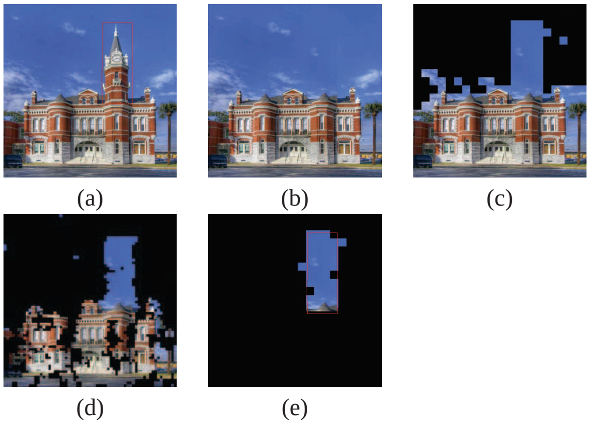


Fig. 13: Comparison for image completed forged images: (a) the original JPEG image, "campanile," (b) the tampered image, (c) the detection result of *MLE* [24], (d) the result of *BAG* approach [25], (e) the result of our approach.

2) Image completion tampering: Figure 13 gives the experiment for image completion tampering. Fig. 13(a) is the original "campanile" image with JPEG block size 8×8. Fig. 13(b) gives the tampered image with the clock tower removed. The removed area is filled in by the state-of-the-art image completion approach proposed in [21]. Fig. 13(c) illustrates the detection results by using the *MLE* approach [24] which results in large portion of false detection due to the interference from strong edges. In Fig. 13(d), the *BAG* approach [25] is also vulnerable to interference from the strong edges, and fails to detect the tampered area precisely. Compared with the *MLE* approach and the *BAG* approach, our proposed approach can effectively detect tampered areas manipulated by image completion.

3) Image composite tampering: Figure 14 shows the detection result for image composite tampering. Fig. 14(a)-(b) show the original target image "battlefield," and the source image, respectively. Fig. 14(c) shows the fake image created by composite tampering which removes the helicopter object in the target image, and completes the remaining hole by image completion algorithm in [21]. For the simulation of the plausible fake image, this study copies and pastes the vehicle as shown in Figure 14(b) in the source image to the completed target image. Figure 14(d) shows the detection result obtained by *MLE* approach. As the image contents contain many edges and texture, it resulted in numerous false detections. Figure 14(e) shows the detection result obtained by the *BAG* approach [25]. The *BAG* approach cannot accurately identify tampered areas because the fake image contains textured areas as well as strong edges. Only our proposed approach can effectively detect and localize the tampering areas, as shown in Fig. 14(f).

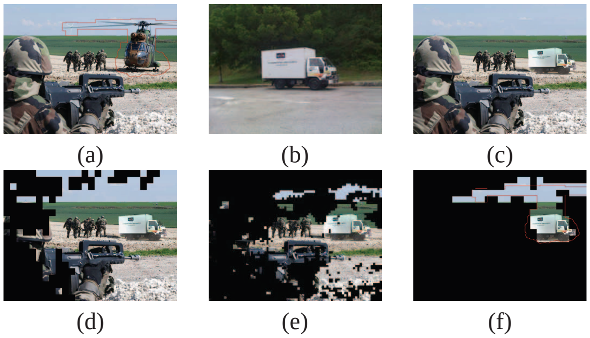


Fig. 14: The composite tampering image: (a) target image, "battlefield;" (b) source image, (c) the fake image, (d) extension of *MLE* [24], (e) *BAG* approach [25], (f) the proposed approach.

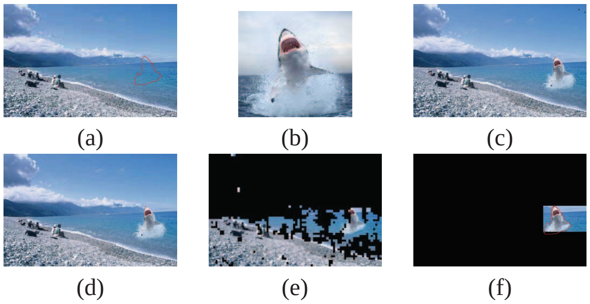


Fig. 15: The copy-and-paste forged by two different JPEG images: (a) target JPEG image with block size 32×32, (b) source JPEG image with block size 8×8, (c) the tampered image, (d) *MLE* approach [24], (e) *BAG* approach [25], (f) our proposed approach.

4) Variable block sizes tampering: Figure 15 shows the result for images with different JPEG block sizes. In Fig. 15(c), a shark copied from a source image with block size 8×8 is pasted to the target image "Beach" which has a smooth region and a high texture region, and block size 32×32. In the experiment, *MLE* fails to estimate the correct primary block size due to interference from high texture, and as shown in Fig. 15(d), almost the whole image is wrongly detected as tampered. In Fig. 15(e), *BAG* also fails in regions with high texture. Our approach is the only one which successfully detects the tampered area as sown in Fig. 15(f).

5.2 Performance and comparison

The performance is measured by the precision, recall, and $F1_$measure in two levels: block level and image level. In block level, we evaluate the accuracy of the detected tampered area; in image level, we focus on that the proposed approach distinguishes whether an image is real or tampered.

The *precision*, *recall*, and $F1\_measure$ are defined as follows. We say that a block, resp. an image, is a *positive* instance if it is tampered; otherwise, it is a *negative* instance. Let TP be the number of true positives, i.e. the number of tampered instances which are detected as tampered, and FP be the number of false positives, i.e. the number of un-tampered instances which are detected as tampered. Similarly, we define FN to be the number of false negatives, and TN to be the number of true negatives. We have the following definitions.

$$Precision(P) = TP/(TP+FP), \tag{12}$$

$$Precision(P) = TP/(TP+FN), \tag{13}$$

$$F1\_measure(F1) = 2 \times P \times R/(P+R). \tag{14}$$

We collect a data set which consists of 800 images, including 20 images downloaded from Internet, and 780 images from UCID database [41]. Images in the data set are randomly partitioned into 3 groups: Original, Tampered I, and Tampered II which consist of 300, 100 and 400 images, respectively. The 300 images in group Original are original, and not tampered. The 100 images in group Tampered I are manipulated manually with copy and paste, image completion [3],[21], image composite tampering [35] and variable block sizes tampering (VBS), respectively. The 400 images in group Tampered II are manipulated by random copy-and-paste as follows. For each image, a randomly selected area from another image with random size is copied and pasted at a random position. The quality factor of JPEG compression ranges from 10 to 95, and the block size includes 8, 16, and 32.

Table 2 gives the block-level results for groups Tampered I and II which are tampered manually and randomly, respectively. The results show that our approach outperforms *MEL* [24] and *BAG* [25] in both manually and randomly tampered images. Both *MEL* and *BAG* produce a large amount of false alarm, and suffer very low precision. Our approach achieves equally well in both high precision and recall, and performs stably for all tampering methods. Similar comparative results in image level are observed in Table 3.

Table 3. Performance in image level

	Evaluation		
Algorithm	P	R	F1
MLE [24]	63.5	93.8	75.8
BAG [25]	62.5	100	76.9
Ours	88.7	100	94.0

5.3 Experiment for parameter selection

In this subsection, we give experimental results for different values of the threshold θ in the proposed enhanced cross difference filter, as well as different threshold values for the ratio of detected tampered blocks to determine tampered blocks in detection result refinement.

Table 4 shows how the performance of *RSVMLE* is affected by various threshold values θ in the proposed enhanced cross difference filter. The 885 test images and the experimental setup are same as in subsection 3.2.3. The results show that the best performance is achieved when θ =15.

Table 5 shows how the performance of refinement process is affected by various threshold values for the ratio of detected blocks to determine tampered blocks. The results show that Recall evaluation decreases as the threshold value increases. This is because larger threshold values will increase the false negative rate. Overall, the value 0.7 gets the best result.

Table 4. The accuracy of *RSVMLE* for various threshold values

	QF		
Threshold	60	75	90
0	99.7	98.1	66.2
15	**100**	**99.7**	**90.1**
35	99.8	99.3	83.1
55	99.8	99.2	77.8

Table 5. Performance of refinement process for different block ratio threshold values

	Threshold		
Evaluation	0.5	0.7	0.9
P	83.3	**83.6**	83.3
R	84.2	**84.3**	82.3
F1	83.0	**83.1**	82.0

6 Conclusion and future remarks

In this paper, we have presented a robust approach for passive forgery detection in JPEG compressed images, which is based on reliable estimation of block sizes from block artifacts resulting from JPEG compression. We have developed an enhanced cross difference filter to produce a map which strengthens block artifacts and reduces interference from strong edges, and integrated techniques from randomly sampling and voting to improve the accuracy of maximum likely estimation. We have carried out experiment to compare our approach with extension of *MLE* [24] and *BAG* [25] over several major tampering methods, including copy-and-paste, image completion and image composite tampering. The experiment shows that our approach can effectively detect and localize tampering areas in all test images, and

Table 1. Characteristics of test images.

Name	Image Size	Block Size	Tampering Method	Content Types
Gold Hill	512×512	8×8	copy-and-paste	intensive edges
Nature scene	539×720	8×8	copy-and-paste	high texture
Campanile	512×512	8×8	image completion	smooth region, strong edge
Battlefield	480×640	8×8	composite tampering	smooth region, strong edge, high texture
Beach	416×640	target: 32×32, source: 8×8.	copy-and-paste	smooth region, high texture

Table 2. Performance in block level

Group	Tampering	Algorithm								
		MLE[24]			BAG [25]			Proposed		
		P	R	F1	P	R	F1	P	R	F1
I	Copy&Paste	28.8	82.1	22.1	19.3	93.5	30.7	85.1	81.9	83.2
	Image completion	34.4	95.4	42.9	28.3	65.2	36.1	91.2	83.9	87.3
	Composite tampering	21.0	93.4	27.4	19.7	76.7	29.8	85.5	81.5	83.1
	VBS tampering	37.8	72.6	29.0	22.3	91.7	34.5	88.6	80.2	83.8
	Group Avg.	30.5	85.6	30.3	22.4	81.8	32.8	87.6	81.9	84.4
II	Random Copy&Paste	24.5	91.8	29.9	26.0	95.6	41.5	83.6	84.3	83.1
Overall Avg.		25.7	90.6	28.4	25.3	92.9	37.9	85.7	83.1	83.8

outperforms *MLE* and *BAG* which result in high false detection rates. In the future, we will continue to improve our approach, and study how to apply the main idea developed in this paper to other problems such as video forgery detection.

References

[1] M. Barni, A. Costanzo, L. Sabatini, Identification of cut & paste pampering by means of double-JPEG detection and image segmentation, in: Proc. of IEEE Int. Conf. on Circuits and Systems, 1687-1690 (2010).

[2] S. Bayram, H.T. Sencar, N. Memon, Source camera identification based on CFA interpolation, in: Proc. of IEEE Int. Conf. on Image Processing, 69-72 (2005).

[3] A. Criminisi, P. Prez, K. Toyama, Region filling and object removal by exemplar-based image inpainting, IEEE Trans. on Image Processing **13**, 1200-1212 (2004).

[4] W. Chen, Y.Q. Shi, W. Su, Image splicing detection using 2-D phase congruency and statistical moments of characteristic function, in: Proc. of SPIE Int. Conf. on Security, Steganography, and Watermarking of Multimedia Contents, 65050R (2007).

[5] Y.L. Chen, C.T. Hsu, Image tampering detection by blocking periodicity analysis in JPEG compressed images, in: Proc. of IEEE Int. Conf. on Multimedia Signal Processing, 803-808 (2008).

[6] Y.L. Chen, C.T. Hsu, Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection, IEEE Trans. on Information Forensics and Security **6**, 396-406 (2011).

[7] M. Chen, J. Fridrich, M. Goljan, J. Lukas, Determining image origin and integrity using sensor noise, IEEE Trans. on Information Forensics and Security **3**, 74-90 (2008).

[8] M. Chen, J. Fridrich, M. Goljan, Defending against fingerprint-copy attack in sensor-based camera identification, IEEE Trans. on Information Forensics and Security **6**, 227-236 (2011).

[9] H. Farid, A survey of image forgery detection, IEEE Signal Processing Magazine **2**, 16-25 (2009).

[10] J. Fridrich, D. Soukal, J. Luks, Detection of copy move forgery in digital images, in: Proc. of Conf. on Digital Forensic Research Workshop, 55-61 (2003).

[11] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM **24**, 381-395 (1981).

[12] Z. Fan, R.L. de Queiroz, Identification of bitmap compression history: JPEG detection and quantizer estimation, IEEE Trans. on Image Processing **12**, 230-235 (2003).

[13] Andrew Walsh, Quick response codes and libraries, Library Hi Tech News, **26**, 7-9 (2009).

[14] J.M. Guo, Y.F. Liu, Z.J. Wu, Duplication forgery detection using improved DAISY descriptor, Expert Systems with Applications **40**, 707-714 (2013).

[15] J.M. Guo, Y.F. Liu, Z.J. Wu, Duplicated forgery detection using improved rotation invariant DAISY descriptor, in: Proc. of IEEE Int. Conf. on Pattern Recognition, 11-15 (2012).

[16] J. Hays, A.A. Efros, Scene completion using millions of photographs, ACM Trans. on Graphic **26**, 1-7 (2007).

[17] A. Imiya, I. Fermin, Motion analysis by random sampling and voting process, Computer Vision and Image Understanding **73**, 309-328 (1999).

[18] M.K. Johnson, H. Farid, Exposing digital forgeries by detecting inconsistencies in lighting, in: Proc. of ACM Int. Conf. on Multimedia and Security Workshop, 1-10 (2005).

[19] M.K. Johnson, H. Farid, Detecting photographic composites of people, in: Proc. of Int. Conf. on Digital Watermarking, 19-33 (2008).

[20] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Prentice-Hall, New Jersey, 2008.

[21] N. Komodakis, G. Tziritas, Image completion using efficient belief propagation via priority scheduling and dynamic pruning, IEEE Trans. on Image Processing **16**, 2649-2661 (2007).

[22] V. Kwatra, A. Schodl, I. Essa, G. Turk, A. Bobick, Graphcut textures: image and video synthesis using graphcuts, ACM Transactions on Graphics **22**, 277-286 (2003).

[23] W. Luo, Z. Qu, J. Huang, G. Qiu, A novel method for detecting cropped and recompressed image blocks, in: Proc. of IEEE Int. Conf. on Acoustic Speech and Signal Processing, 217-220 (2007).

[24] W.S. Lin, S.K. Tjoa, H.V. Zhao, K.J. Ray Liu, Digital image source coder forensics via intrinsic fingerprints, IEEE Trans. on Information Forensics and Security **4**, 460-475 (2009).

[25] W. Li, Y. Yuan, N. Yu, Passive detection of doctored JPEG image via block artifact grid extraction, Signal Processing **89**, 1821-1829 (2009).

[26] Z. Lin, J. He, X. Tang, C.K. Tang, Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis, Pattern Recognition **42**, 2492-2501 (2009).

[27] J. Lukas, J. Fridrich, Estimation of primary quantization matrix in double compressed JPEG images, in: Proc. of Int. Conf. on Digital Forensic Research Workshop, pp. 5-8 (2003).

[28] G.S. Lin, M.K. Chang, Y.L. Chen, A passive blind forgery detection scheme based on content-adaptive quantization table estimation, IEEE Trans. on Circuits and Systems for Video Technology **21**, 421-434 (2011).

[29] Z. Lin, R. Wang, X. Tang, H.Y. Shum, Detecting doctored images using camera response normality and consistency, in: Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition, 1087-1092 (2005).

[30] Q. Liu, X. Cao, C. Deng, X. Guo, Identifying image composites through shadow matte consistency, IEEE Trans. on Information Forensics and Security **6**, 1111-1122 (2011).

[31] G. Voyatzis, I. Pitas, Protecting digital image copyrights: a framework, Computer Graphics and Applications, IEEE, **19**, 18-24 (1999).

[32] B. Mahdian, S. Saic, A bibliography on blind methods for identifying image forgery, Signal Processing: Image Communication **25**, 389-399 (2010).

[33] T.T. Ng, S.F. Chang, Q. Sun, Blind detection of photomontage using higher order statistics, in: Proc. of IEEE Int. Symposium on Circuits and Systems, 688-691 (2004).

[34] T.T. Ng, S.F. Chang, A model for image splicing, in: Proc. of IEEE Int. Conf. on Image Processing, 1169-1172 (2004).

[35] P. Prez, M. Gangnet, A. Blake, Poisson image editing, ACM Trans. on Graphics **22**, 313-318 (2003).

[36] A.C. Popescu, H. Farid, Statistical tools for digital forensics, in: Proc. of Int. Conf. on Information Hiding, 128-147 (2004).

[37] A.C. Popescu, H. Farid, Exposing digital forgeries by detecting traces of re-sampling, IEEE Trans. on Signal Processing **53**, 758-767 (2005).

[38] T. Pevny, J. Fridrich, Detection of double-compression in JPEG images for applications in steganography, IEEE Trans. on Information Forensics and Security **3**, 247-258 (2008).

[39] A.C. Popescu, H. Farid, Exposing digital forgeries in color filter array interpolated images, IEEE Trans. on Signal Processing **53**, 3948-3959 (2005).

[40] W.B. Pennebaker, J.L. Mitchell, JPEG Still Image Data Compression Standard, Van Nostrand Reinhold, New York, 1993.

[41] G. Schaefer, M. Stich, UCID-an uncompressed color image database, in: Proc. of SPIE Electronic Imaging, Storage and Retrieval Methods and Applications for Multimedia, 472-480 (2004).

[42] J. Wise, J. Caprio, T. Parks, Maximum likelihood pitch estimation, IEEE Trans. on Acoust., Speech, Signal Processing **24**, 418-423 (1976).

[43] S. Ye, Q. Sun, E.C. Chang, Detecting digital image forgeries by measuring inconsistencies of blocking artifact, in: Proc. of IEEE Int. Conf. on Multimedia Expo., 12-15 (2007).

[44] W. Zhang, X. Cao, Z. Feng, J. Zhang, P. Wang, Detecting photographic composites using two-view geometrical constraints, in: Proc. of IEEE Int. Conf. on Multimedia and Expo., 1078-1081 (2009).

[45] JPEG club, http://jpegclub.org/djpeg/.

[46] Department of Computer Science and Information Engineering, National Chung Cheng University, Taiwan, https://sites.google.com/site/imageforgerydetection1/.

Cheng-Shian Lin received the M.B.A. degree in Information Management from Hsuan Chuang University, Taiwan, R.O.C. in 2005. He is currently a Ph.D. in the Graduate Institute of Computer Science and Information Engineering, National Chung Cheng University, Taiwan, R.O.C. His research interests include computer vision, image authentication, and image/video forensics.

Jyh-Jong Tsay is an Associate Professor at the Department of Computer Science and Information Engineering at National Chung Cheng University, Chiayi, Taiwan, Republic of China. He completed his PhD Degree at Purdue University, USA, in 1990. His research interests include machine learning, data mining, information retrieval and image/video forensics.