Detection of Copy-Move Forgery in Digital Images Using SIFT Algorithm

Hailing Huang[1], Weiqiang Guo[1], Yu Zhang[1]
[1]*Research Institution of Computer Application,*
South China University of Technology,Guangzhou, 510640,China
helly505@sina.com, {wqguo, yzhang}@cellcom.com.cn

Abstract

As result of powerful image processing tools, digital image forgeries have already become a serious social problem. In this paper we describe an effective method to detect Copy-Move forgery in digital images. This method works by first extracting SIFT descriptors of an image, which are invariant to changes in illumination, rotation, scaling etc. Owing to the similarity between pasted region and copied region, descriptors are then matched between each other to seek for any possible forgery in images. Experiments have been performed to demonstrate the efficiency of this method on different forgeries and quantify its robustness and sensitivity to post image processing, such as additive noise and lossy JPEG compression etc, or even compound processing.

1. Introduction

Due to the widespread use of sophisticated imaging tools, it is increasingly easier to manipulate digital images and create forgeries that are difficult to distinguish from authentic photographs. However, abusive use of digital forgeries has become a serious problem, in various fields like medical imaging, digital forensics, journalism, scientific publications, etc. Over the past few years, different approaches have been proposed to identify digital images, to confirm image contents and detect their tampering. One method is to embed digital watermarks or signatures in digital images in advance [1]. But this method decreases image quality and security, and moreover, it is limited to controlled environments, where images are created with digital watermarks or signatures ahead. The other method is passive-blind authentication [2], which has become a new research focus in the multimedia security field. Without relying on any pre-extraction or pre-embedded information, passive blind technology only makes use of characteristics of images, thus it is more flexible and practical actually.

A special type of digital forgery with images is Copy-Move forgery or Region-Duplication forgery, in which part of images is copied and pasted somewhere else in the identical images to conceal a person or object in the scene intentionally. In this paper, we present an efficient method to detect and localize such forgery in digital images with no pre-extraction or pre-embedded information.

The rest of the paper is organized as follows. A detailed detection method is provided in section 2. And section 3 presents the results of the experiment to show the effectiveness of the proposed method. Finally, we conclude in Section 4.

2. Detection of Copy-Move Forgery

Copy-Move forgery is one of the common image forgery techniques. The attackers copy a part of an image itself and, with the purpose of disguising some details, paste it to another part of the same image. Unlike coping from different source images, coping from the same image has more chance of leaving behind no traces of tampering, for the reason that, attributes in the same image, such as illumination, proportion, or focus etc, are resemble. Furthermore, some attackers may perform some post processing attack after Copy-Move operation, which makes the task of detecting forgery significantly harder. Hence, the key of detection method is the robustness against the post image processing, such as noise contamination, lossy JPEG compression, blurring etc.

Since Copy-Move forgery imports a correlation between the original image region and the pasted region, it is feasible to take advantage of this correlation as a basis for successful detection of a forgery by looking for identical image regions [3]. Therefore we propose an effective detection method using feature matching technology to authenticate duplication regions in forgery images. In order to improve the precision of detection and robustness against the post image processing, we introduce SIFT (Scale Invariant Feature Transform) algorithm to our method, whose strong matching ability and stability in noise, rotation and variety of different scale have made it one of the most successful algorithms in feature-based matching research in recent years. Now we give the proposed detection method based on SIFT below.

2.1. Generating the image features

978-0-7695-3490-9/08 $25.00 © 2008 IEEE
DOI 10.1109/PACIIA.2008.240

The SIFT algorithm extracts distinctive features of local image patches which are invariant to image scale and rotation and are robust to changes in noise, illumination, distortion and viewpoint. As described in [4], it consists of four major steps: (1) Scale-space extrema detection; (2) Keypoint localization; (3) Orientation assignment; (4)Keypoint descriptor.

2.1.1 Scale-space extrema detection. The first step of the computation searches for extrema over all scales and image locations. Given an input image $I(x,y)$, then the scale space of image I is defined as follows:

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \tag{1}$$

where * is the convolution operation in x and y directions, and the Gaussian function

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{\sigma 2}} \tag{2}$$

where σ is the factor of scale space. In order to efficiently detect potential interest points that are invariant to scale and orientation, which are also called keypoints in SIFT framework, the method used the scale-space extrema in the difference-of-Gaussian(DoG) function convolved with the image, $D(x,y,\sigma)$, which can be computed from the difference of two nearby scales separated by a constant multiplicative factor k:

$$\begin{aligned} D(x,y,\sigma) &= [G(x,y,k\sigma) - G(x,y,\sigma)] * I(x,y) \\ &= L(x,y,k\sigma) - L(x,y,\sigma) \end{aligned} \tag{3}$$

The convolved images are grouped by octave, and an octave corresponds to doubling the value of σ. Then the value of k is selected so that we obtain a fixed number of blurred images per octave. This also ensures the same numbers of DoG images are generated per octave. Once DoG images have been obtained, keypoints are identified as local minima or maxima of the DoG images across scales. This is done by comparing each pixel in the DoG images to its eight neighbors at the same scale and nine corresponding neighboring pixels in each of the neighboring scales. If the pixel value is the maximum or minimum among all compared pixels, it is selected as a candidate keypoint.

2.2.2 Keypoint localization. Scale-space extrema detection produces too many keypoint candidates, some of which are unstable. Then keypoints are filtered in this step so that only stable keypoints are retained. Once a keypoint candidate has been found by comparing a pixel to its neighbors, next is to perform a detailed fit to the nearby data for accurate location, scale, and ratio of principal curvatures. This information allows points to be rejected that have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge. Details are given in [4].

2.2.3 Orientation assignment. This is the key step to achieve invariance to image rotation, in which each keypoint is assigned one or more orientations based on local image gradient directions. The keypoint orientation is calculated from an orientation histogram of local gradients from the closest smoothed image $L(x,y,\sigma)$. For each image sample $L(x,y)$ at the keypoint's scale σ, the gradient magnitude $m(x,y)$ and orientation $\theta(x,y)$ is computed using pixel differences, let

$$m(x,y) = \sqrt{L_1^2 + L_2^2} \tag{4}$$

$$\theta(x,y) = \arctan(L_2 / L_1) \tag{5}$$

where $L_1 = L(x+1,y,\sigma) - L(x-1,y,\sigma)$, and $L_2 = L(x,y+1,\sigma) - L(x,y-1,\sigma)$.

An orientation histogram with 36 bins, with each bin covering 10 degrees, is formed from the gradient orientations of sample points within a region around the keypoint. Then the maximum orientation is assigned to this keypoint; additional keypoints will be created with orientation which is within 80% of the maximum orientation.

2.2.4 Keypoint descriptor. The previous operations have assigned an image location, scale, and orientation to each keypoint, which ensure the invariance to image rotation, location and scale. And then we want to compute descriptor vectors for each keypoint such that descriptors are distinctive and robust to other variations, such as illuminations etc. Compute the feature descriptor as a set of orientation histograms on 4 x 4 pixel neighborhoods. The orientation histograms are relative to the keypoint orientation and the orientation data comes from the Gaussian image closest in scale to the keypoint's scale. Histograms contain 8 bins each, and each descriptor contains a 4x4 array of 16 histograms around the keypoint. This leads to a SIFT feature vector with (4 x 4 x 8 =) 128 elements.

2.2. Detecting forgeries by feature matching

After extracting SIFT keypoints from an input unknown image, these distinctive keypoints are then matched between each other to authenticate copy-move forgeries in the digital image. If any matches are detected, it means that the inputted image has copy-move forgeries. And the matching result will be displayed in the image to see which part of the image is covered.

In our paper, the match for each keypoint is found by identifying its nearest neighbor, which is defined as the keypoint with minimum Euclidean distance for descriptor vector. To increase robustness, matches are accepted only if the ratio of closest to second-closest neighbors is less than a threshold ω. And if this distance ratio threshold is decreased, the number of matches of keypoints would

drop accordingly, but the accuracy of match would get enhanced. Thus, appropriate threshold is preferable for reducing number of incorrect matches, which will be further discussed in Section 3.

However, it is known to have high complexity to identify the most similar vectors from many high dimensional vectors. Fortunately, the BBF (Best-Bin-First) search method [5] derived from the k-d tree algorithm can identify the nearest neighbors with high probability using only a limited amount of computation. Although lowering space dimensions of vectors is another feasible way, we still set the vectors as 128 elements to make the detection result more accurate.

But in traditional searching process, there are often two sets of keypoints, extracted from two images needed to be matched, with one set to generate k-d tree, and with another to search value in the tree. And in our task, the inputted image just generate only one set of keypoints in the previous steps. Hence, we use the following searching strategy:

(1) Given a set of generated feature keypoints as $\varnothing K$, then divide this set into two sets as $\varnothing K_1$, $\varnothing K_2$ .If $p = [size(\varnothing K)/2]$, then $\varnothing K_1$ is the first p elements of $\varnothing K$, and $\varnothing K_2$ is the remaining elements.

(2)Using $\varnothing K_2$ and $\varnothing K_1$ to BBF search method, and then save the matching result.

(3)Divide $\varnothing K_1$ and $\varnothing K_2$ respectively like (1) and repeat (1) (2) until every keypoint in $\varnothing K$ is matched with each other.

We have implemented the detection method in C++, and the experimental environment is on a personal computer of 2.2GHz processors with 2GB memory. Tests have been performed on various images collected from the internet, considering different sized duplication region and different post image processing.

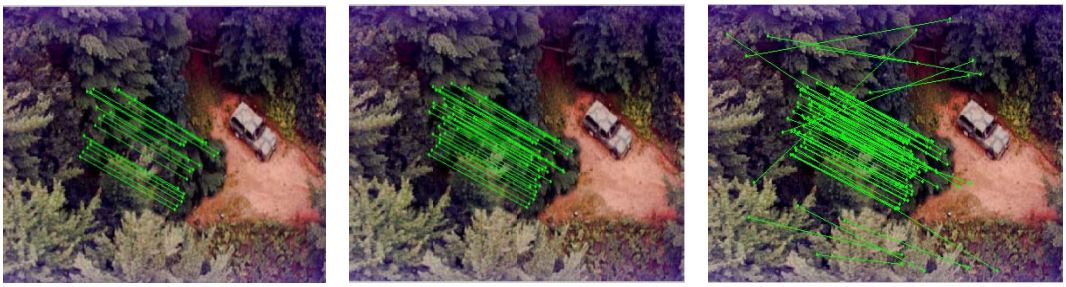
As shown in Figure 1, (a) is an original image and (b) is the forged imaged of (a). When compared (a) with (b), we can find little traces of tampering, in which a jeep was covered with a portion of the foliage left, because the foliage texture makes it too difficult to identify the forged region visually. Use our detection method to detect forgeries in Figure 1(b). SIFT features keypoints are extracted first, and next they are matched within an appropriate threshold ω. Then the result of detection is displayed in the image with lines between two matched keypoints. Knowing the detection output, we can visually confirm the duplication region and discard some mismatched points in the tampered image.

The comparison of threshold ω in different value is given in Figure 2. From Figure 2, we can see that, higher threshold value may bring about more false matching, while lower threshold value could miss some correct matching. Thus accurate detection needs appropriate threshold, which can be obtained through several times of tests. In Table 1 we list the matching result of Figure 1(b) with more threshold value from 0 to 1(not included). When the value exceeds 0.7, false matching is obvious. From our experimental results, when the value ranges between 0.3 and 0.55, the detection result is more acceptable and accurate. In this paper, we set this threshold value as 0.45.

3. Experimental Results

(a) Original image **(b) Tampered image**

Figure 1 Tampered jeep image with its original image



(a) ω=0.3 (b) ω=0.45 (c) ω=0.6

Figure 2 Detection result of tampered jeep image

Table 1 Matching result with different threshold value

Threshold value	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Number of matches	0	6	19	38	57	80	140	340	949
Number of false matches	0	0	0	0	2	12	too many	too many	too many

To evaluate the robustness and sensitivity of our method, we perform some common post image processing to tampered images in the constructed database, which contains 100 images of different size. Most of the images are from the internet or papers. Each image in the database is either rotated in different angles, or JPEG compressed with various quality factors, or corrupted with additive noise with varying signal to noise ratios (SNR), or Gaussian blurred etc. Figure 4~6 shows the output of our proposed detection method applied to various tampered example images with a threshold $\omega = 0.45$.

Shown in Figure 4, (a) is of size 256x256, with 90 degree rotation; (b) is of size 256x256, with 180 degree rotation; (c) is of size 460x460, with 270 degree rotation. Although there are two incorrect matches in (a), the method in general is robust to different rotation with different size of copied region.

Figure 3 is another forged gray image example used by [6]. (a) is the original image and (b) is the tampered image. From detection output, we can see that the mountain background is used to conceal the left person without any traces. Shown in Figure 5, are images corrupted with different SNR of additive white Gaussian noise. Figure 6 presents images compressed with different JPEG quality.

Generally, our proposed method is of great accuracy, except for very small block sizes. Note that the average number of false positives (regions incorrectly labeled as duplicated) is relatively low. Experiments on the constructed database show that proposed method is accurate and is robust to many post image processing, such as rotation, noise, JPEG compression. Especially it is also effective to compound image processing, for example, first rotation and then JPEG compression. However, the false detection may increase when the SNR is lower than 20dB or quality factors of JPEG compression is lower than 40.

4. Conclusions

As a common forgery operation, Copy-Move forgery uses a part of an image to conceal another part of the same image. Taking advantage of similar texture characteristic in the same image, especially after some image processing, forged images are difficult to be identified visually. Compared with other forgery technologies, although it is very easy to manipulate, it also has its own shortage: correlation between copied part and pasted part. Based on such fact, we propose an effective and robust method based on feature matching to detect Copy-Move forgery in different kinds of digital images. Due to the strong stability of SIFT feature descriptors, our method has a good performance on different kind of post image processing (JPEG compression, rotation, noise, scaling etc), and is also robust to compound image processing. However, further investigation is still needed to improve the robustness against low SNR and small size tampered region.

References

[1] J. Fridrich (1999). "Methods for Tamper Detection in Digital Images". In *Proc. of ACM Workshop on Multimedia and Security*, Orlando, FL, pp.19-23.

[2] T. T. Ng, S. F. Chang, Q.B. Sun (2004). "Blind Detection of Photomontage Using Higher Order Statistics". *IEEE International Symposium on Circuits and Systems (ISCAS)*, Vancouver, Canada, Vol.5, pp.688–691.

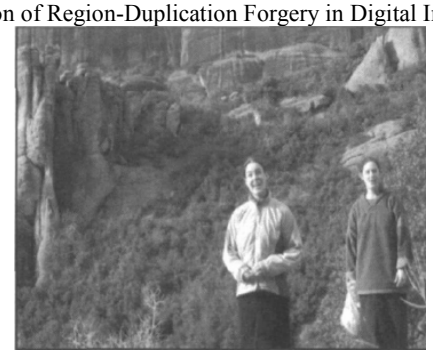
[3] A.N. Myrna, M.G. Venkateshmurthy, C.G. Patil (2007). "Detection of Region Duplication Forgery in Digital Images Using Wavelets and Log-Polar Mapping". *IEEE Conference on Computational Intelligence and Multimedia Applications*, Vol. 3, pp.371~377.

[4] D.G. Lowe (2004). "Distinctive Image Features from Scale-Invariant Keypoints". *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110.

[5] J. S. Beis, D.G. Lowe (1997). "Shape Indexing Using Approximate Nearest-Neighbour Search in High Dimensional Spaces". *IEEE Conference on Computer*

Vision and Pattern Recognition, Puerto Rico, pp. 1000-1006.

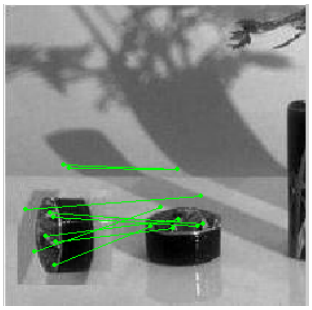
[6] W.Q. Luo, J.W. Huang, G.P. Qiu (2006). "Robust Detection of Region-Duplication Forgery in Digital Image". In *Proc. of 18th International Conference on Pattern Recognition*, Hong Kong, Vol.4, pp. 746-749.



(a) Original image

(b) Tampered image

Figure 3 Gray image example



(a) θ=90°

(b) θ=180°

(c) θ=270°

Figure 4 Detection of tampered images with different rotation

(a)SNR (20dB)

(b) SNR (30dB)

Figure 5 Detection of different white Gaussian noise

(a)JPEG 40

(b) JPEG 65

Figure 6 Detection of different JPEG compression