A Study on Features for the Detection of Copy-Move Forgeries

Vincent Christlein, Christian Riess, Elli Angelopoulou

{sivichri@stud, riess@i5, elli@i5}.informatik.uni-erlangen.de

Abstract: Blind image forensics aims to assess image authenticity. One of the most popular families of methods from this category is the detection of copy-move forgeries. In the past years, more than 15 different algorithms have been proposed for copy-move forgery detection. So far, the efficacy of these approaches has barely been examined. In this paper, we: a) present a common pipeline for copy-move forgery detection, b) perform a comparative study on 10 proposed copy-move features and c) introduce a new benchmark database for copy-move forgery detection. Experiments show that the recently proposed Fourier-Mellin features perform outstandingly if no geometric transformations are applied to the copied region. Furthermore, our experiments strongly support the use of kd-trees for the matching of similar blocks instead of lexicographic sorting.

1 Introduction

The goal of blind image forensics is to examine image authenticity without the help of an embedded security scheme (like watermarks). In the past few years, a diverse set of methods for blind image forensics has been developed, see e. g. [SM08, NCLS06, Far09a]. The existing methods can be divided into three categories. The first group focuses on detecting the existence of expected artifacts that have been introduced in the process of image sensing. The detection of inconsistensies in such artifacts is a cue for the lack of authenticity of an image. Particular examples include the recovery of a camera-specific sensor noise pattern by Lukas *et al.* [LFG06], the estimation of the camera response function [HC07], demosaicing [GC08], or the extraction of lens properties [JF06].

A second approach is to search for inconsistencies in the scene that have been introduced as a side effect of tampering. Johnson and Farid presented an approach for the examination of the illumination direction [JF05] and for the geometric composition of the light sources in a scene [JF07]. Others like Lalonde and Efros used color distributions in images in order to detect spliced images [LE07].

Finally, many groups considered image artifacts that are introduced by the tampering operations themselves. Examples are artifacts from double JPEG compression (e. g. [HLWT06, Far09b]) or the detection of resampling artifacts [PF05]. One of the most actively researched topics in this area is the detection of copy-move forgeries, see for example [LHQ06, BSN09, LWK09, MS07, KW08, BSM09]. Copy-move forgery detection (CMFD) aims at finding regions that have been copied and pasted within the same image. The goal of such a tampering operation is to either hide unwanted parts of the image, or to emphasize particular image content, like enlarging a crowd of people. Figure 1 shows an example that was originally presented in [MS07]. In this case, grass has been copied over one of the soldiers.

Unfortunately, although a considerable number of solutions have been proposed for this problem, a thorough comparison of these methods has not been performed so far. We believe that there are two reasons for the lack of such a comparative evaluation: the missing common ground truth data (i. e. standardized benchmark datasets) and the lack of standard metrics for their evaluation.

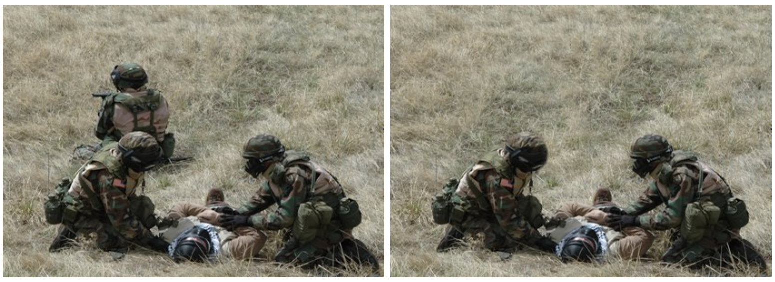


Figure 1: Example for copy-move forgery from [MS07] (original image left, tampered image right). Grass has been copied to hide one of the soldiers.

In this work, we perform a comparative study on a subset of different feature sets that have been previously proposed. To accomplish this, we: a) formulate existing CMFD methods as concrete instances of a more abstract "copy-move detection pipeline", b) perform a comparison of the methods within this pipeline, and c) present a diverse set of benchmark images for the evaluation of these methods. We believe that this comparison gives further insights into the strengths and weaknesses of particular features for copy-move detection. Furthermore, we propose a common metric for benchmarking copy-move detection methods.

2 Database and Testing Protocol

To create the dataset, we defined 48 manipulation cases. The manipulations were performed by three individuals with different technical skills, so that each person worked on 16 images. Figure 2 and Figure 3 present selected examples from the database. The database is available on our web page[1]. The core manipulation technique that was performed is copy-move forgery. Hence, the major part of every manipulated region stems from another region of the same image. For a more realistic forgery, the boundaries of the copied regions were adjusted with standard image editing methods (like partial trans-

[1] `http://www5.informatik.uni-erlangen.de/data`

parency, painting and smearing). Finally, per-pixel labels are automatically assigned by a supplementary software. The assigned labels are “original”, “altered” and “copy-moved”, as shown in Figure 3. For this study, only “original” and “copy-moved” pixels were considered (see below).

2.1 Source Manipulation Properties

We chose six images from the database (originating from different camera types) to compare the features of ten copy-move forgery detection methods. The images have been selected so that they contain representative challenges for copy-move detection algorithms (see also Sect. 4). They contain different numbers of manipulations, as real-world forgeries. The image size is very important for the detection algorithms. Thus, three of the images have a relatively high resolution of more than 2000×1600 pixels, while the other three images have a lower resolution of about 1000×800 pixels. The duplicated regions themselves also vary significantly in size. Thus, feature sets that make assumptions about the size of the manipulated region are at a disadvantage.

The boundaries of the tampered regions are additionally postprocessed with standard image editing methods. The desired effect is that no clear edges appear and the region fits better with its surroundings. Another way is to create shadows, so that the lightning appears correct. This is a common operation in real-life forgeries in order to disguise the copied region. Three of the six images used in this paper are shown in Figure 2. As can be seen, the operations performed in these images vary in type and size of the copied area. For instance, the neck markings of the giraffe and a modified grass patch affect very small regions, while the statues at acropolis involve entire landmarks.

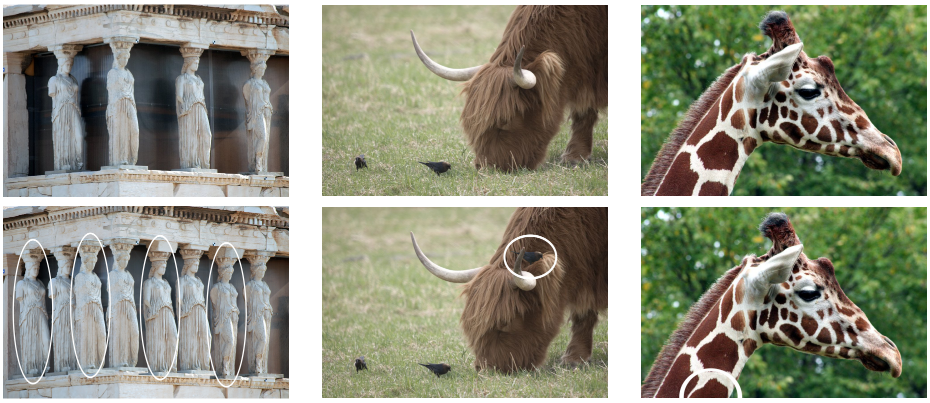


Figure 2: Original images are in the top row, forgeries are in the bottom row. From left to right: *Acropolis* (large copied region), *Cattle* (small region), *Giraffe* (small region)

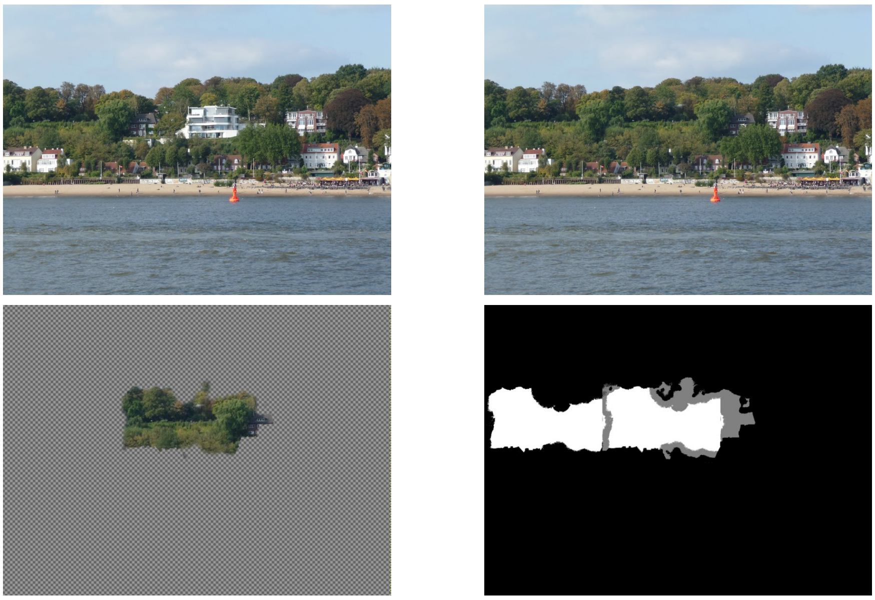


Figure 3: The image *Beachwood* (upper left) is forged with a green patch (bottom left) to conceal a building (upper right). A ground truth map (bottom right) is generated where copy-moved pixels are white, unaltered pixels are black and boundary pixels are gray.

2.2 Testing Protocol

We created ground truth labels for every image. In the ground truth image, the postprocessed boundaries of the duplicated regions are marked differently than the exactly copied pixels (see Figure 3). In our evaluation, we only use exactly copied pixels. Though many methods claim that they can detect such postprocessed parts, we chose to exclude them for two reasons:

1. Most copy-move forgery detection (CMFD) methods mark blocks of copied pixels. Thus, a block on the boundary between copied and non-copied pixels is not well defined as "copied" or "original".
2. One can not clearly define a set of permissible boundary manipulations, in such a way that these pixels can still be considered as copy-moved.

Hence, we believe that the cleanest solution is to exclude boundary pixels from the evaluation. Note that, although not included in this study, this definition does not hinder us to apply "global" post-processing on the image like additive noise or JPEG compression. The robustness of the methods against various kinds of postprocessing can still be evaluated by introducing these artifacts on the copy-moved regions in a controlled manner.

2.3 Detection Error Measures

We employed two basic error measurements, following the ideas of [BSN09] and [LHQ06]. The percentage of false positive pixels F_P and the false negative rate F_N. More precisely, let R_1 be the copied region, R_i, $i > 1$ be the i pasted regions and B the unchanged background. Then,

$$F_\mathrm{P} = \frac{|\text{matches in } B|}{|B|}$$

and

$$F_\mathrm{N} = \frac{|\text{missed matches in } (\bigcup_i R_i)|}{|\bigcup_i R_i|} \quad ,$$

so that lower rates of F_N and F_P indicate higher accuracy.

Note that, as long as a copied region is detected, we consider a high F_P rate to be worse than a high F_N rate. The reason is that high F_P rates lead to a highly confusing overdetection result, which: a) enforces a man-in-the-loop to examine every result and b) might even conceal the truly tampered regions. Also note that the proposed metrics are only applicable in pixel- or block-based methods. In this work, we considered only block-based methods, so that this metric is appropriate. To compare keypoint based methods like [HGZ08] with each other, the absolute number of false positives and false negatives could be used as an approximate indicator of the algorithm performance.

3 Methods and settings

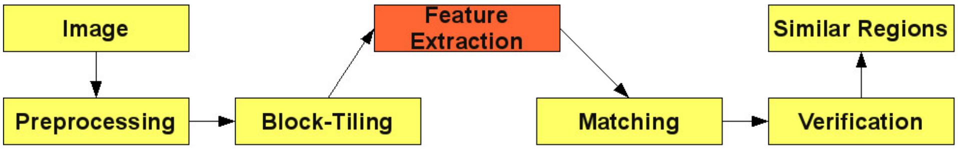


Figure 4: Pipeline of copy-move forgery detection algorithms

The known methods for block-based CMFD follow a similar structure. We built a common framework for CMFD methods that adhere to this structure. Figure 4 shows an overview of the algorithm pipeline containing the main processing steps. After a possible preprocessing step, the image is tiled in small overlapping blocks. For every block a discriminating feature vector is computed. The main difference between the various methods lies in feature extraction.

Similar blocks are pairwise grouped in the matching step. The matching itself is most often done with so-called “lexicographic sorting”. Feature vectors are put row-wise in a matrix and (row-wise) lexicographically sorted. Assuming that similar blocks yield similar features, sufficiently close (w.r.t. a distance measure on the feature vector space) consecutive rows are matched as pairs. Another, not as widely used, approach is to use a kd-tree to find the nearest neighbor in the d-dimensional feature space directly [MS07]. A block pair is typically subject to further constraints. Most often, two matching blocks must have a minimal Euclidean distance between them. This should avoid matching blocks within the same (homogeneous) region.

The verification step filters matching pairs that follow a common pattern. The most common criterion for the verification step is the recognition of same-shift-vectors. We also use this approach in our evaluation. We sort the found matches by their shift-vectors. All shift-vectors that occur more often than a threshold N_f are assumed to be tampered. Note that this approach is inherently limited when geometric transformations, like rotation and scaling, are applied to the copied regions.

The most distinguishing property of the various CMFD algorithms is the employed feature vector. Features of ten copy-move forgery detection algorithms were evaluated. These features can be divided in four groups: moment-based, dimensionality reduction-based, color-based, and frequency domain-based features. MOMENTS 1 denotes the approach of [MS07]. It uses 24 blur-invariant moments as features. MOMENTS 2 uses the first four Hu-moments as features [WLZ$^+$09]. In [PF04], the feature matching space is reduced with principal component analysis (PCA). [KW08] computes the singular values of a reduced-rank approximation (SVD). A third approach using discrete wavelet transformation (DWT) and Singular Value Decomposition [LWTS07] did not give reliable results in our setup and thus is excluded from the evaluation. The first three features used in [LHQ06] and [BSN09] are the average red, green and blue components. Additionally, Luo et al. [LHQ06] use directional information of blocks (COLOR 1) while Bravo-Solorio et al. [BSN09] consider the entropy of a block as discriminating feature (COLOR 2). COLOR 3 by [LWK09] computes the average grayscale intensities of a block and its sub-blocks. The last three methods we investigated consider features in the frequency domain. The features in [FSL03] are 256 coefficients of the discrete cosine transformation (DCT). Coefficients of a discrete wavelet transform (DWT) using Haar-Wavelets form the features in [BNO$^+$07]. A Fourier-Mellin Transform (FMT) yields the features in [BSM09].

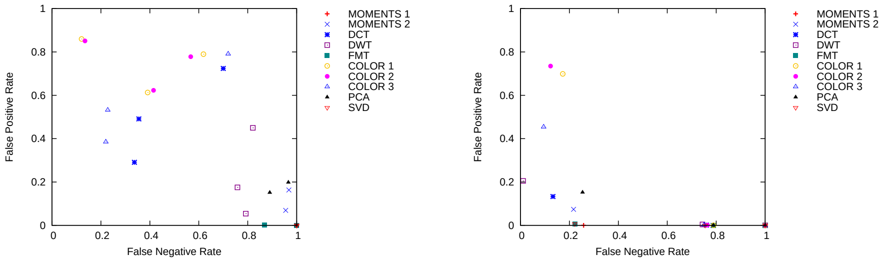
A uniform parameter setting was chosen for better comparison of the individual features. Most of the methods use a block size of 16×16 pixels because it provides a good trade-off between robustness and false positives. We fixed the block size for tiling the image in overlapping blocks accordingly to 16×16 pixels. The step size between every two neighboring blocks was set to 2 pixels to generate fewer blocks and thus decrease the computational complexity. No further preprocessing except grayscale conversion of the image was applied (for algorithms that assume grayscale images).

In order to test the robustness of the features, only the feature sets were exchanged during our different evaluation runs. To decrease the false positive rate, the minimum frequency threshold for the numbers of same-shift-vectors was set to $N_f = 50$. Thus, the minimal area that can be detected is of size 31×31 pixels. The minimum distance between matching blocks was set to 50 pixels, which is also supported by the benchmark images set.

4 Evaluation

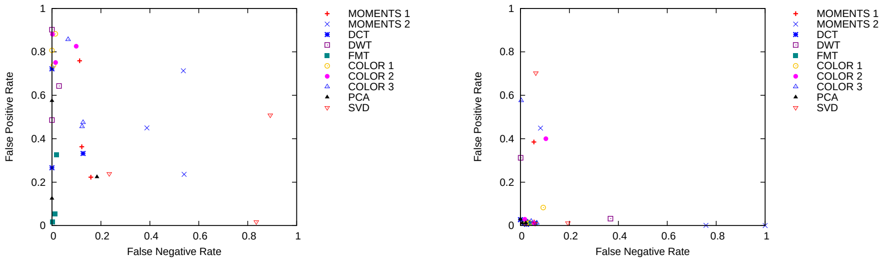
We divided the benchmark images in two groups, large images and small images. Large images are inherently more challenging, since an overall higher number of feature vectors exists, and thus there is a considerably higher probability of matching wrong blocks. On the large image set when using lexicographic sorting, only color features and DCT were able to detect duplicated regions with acceptable error rates, see Figure 5a. We observed that the weakest spot of the other methods is that if too many blocks are wrongly paired, this indeed leads to wrong hypotheses on the dominant shift-vectors, rendering the result barely usable.

Note the comparably weak performance of MOMENTS 1 and MOMENTS 2, on large images as well as on small images. MOMENTS2 was able to detect forgeries in only one small image, MOMENTS 1 always had a false negative rate of 1 and could, hence, detect nothing (Figure 5b).



(a) Large images (b) Small images

Figure 5: Feature Test – Lexicographic sort



(a) Large images (b) Small images

Figure 6: Feature Test – kd-tree sorting

Surprisingly, this behavior completely changes when using a kd-tree. Here, the moment features detect manipulations in all images. In large images (Figure 6a), only FMT and

DCT yield better detection results than MOMENTS 1. PCA gives reliable results, too, while SVD detects duplicated regions in only one large image. Among the color-based methods, COLOR 3 performs best. The features of COLOR 1 and COLOR 2 have very high false positive rates and can not be reliably used for detection.

We assume that this behavior comes from the nature of lexicographic sorting. If the first few elements of the feature vector are indeed the most significant entries, lexicographic sorting might be an appropriate choice. Apparently, the more balanced distance computation of the kd-tree supports the nature of most feature vectors better. On the other hand, many feature vectors exhibit an oversensitivity when using a kd-tree, leading to extremely high F_{P} rates (Figure 6a).

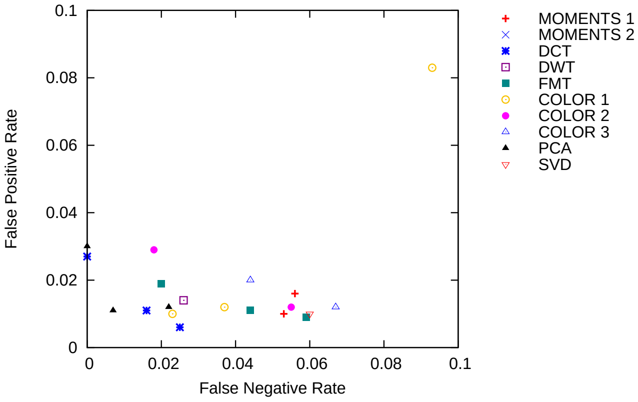


Figure 7: Feature Test – kd-tree representation with small images $[0, 0.1]$

For small images, nearly every feature can be used to detect copy-move forgeries with a kd-tree (Figure 6b). Only MOMENTS 2 and COLOR 3 showed difficulties in a single small image. Upon closer examination, one can notice that the error rates of the small images exhibit a very low false positive rate for all methods (Figure 7). Tables 1b and 1a show the overall average error rates. For the features, FMT has the lowest and, hence, best average error rates when a kd-tree representation is used. On average the F_{P} rate is lower with lexicographic sort. Conversely, the F_{N} rate is typically remarkably better when using a kd-tree representation.

4.1 Geometric Transformation

We also tested the CMFD features on two geometric transformations: scaling and rotation. The copied region in the *Tree* image was rotated up to 20° and scaled up to 120% (see Figure 8). The methods SVD, MOMENTS 2 and COLOR 3 could not be tested against geometric transformations due to their high error rates on the untransformed images. We computed the presented results with a kd-tree, since we observed that lexicographic sorting is too sensitive to the transformations and therefore did not produce reliable results.

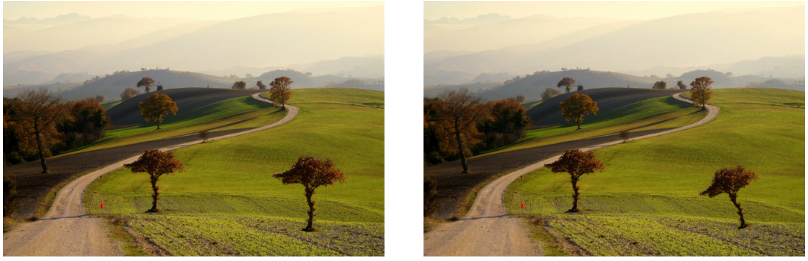
	F_N	F_P	avg
Color 1	0.515	0.494	0.505
Color 2	0.501	0.498	0.499
Color 3	0.502	0.361	0.431
DCT	0.547	0.273	0.410
DWT	0.688	0.148	0.418
FMT	0.813	0.001	0.407
Moments 1	0.841	0.000	0.421
Moments 2	0.821	0.051	0.436
PCA	0.816	0.084	0.450
SVD	0.830	0.001	0.415
	0.687	0.191	0.439

(a) Lexicographic sort

	F_N	F_P	avg
Color 1	0.029	0.422	0.225
Color 2	0.049	0.483	0.266
Color 3	0.072	0.399	0.235
DCT	0.028	0.227	0.128
DWT	0.071	0.398	0.234
FMT	0.026	0.073	0.050
Moments 1	0.093	0.293	0.193
Moments 2	0.551	0.308	0.429
PCA	0.035	0.163	0.099
SVD	0.380	0.248	0.314
	0.133	0.302	0.217

(b) Kd-tree sorting

Table 1: Overall error rates of the feature test with lexicographic sort on the left side and kd-tree representation on the right side, avg = $(F_N + F_P)/2$.

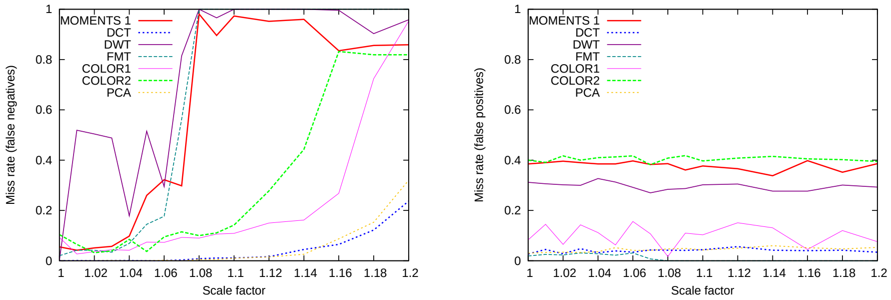


(a) Scaling with 120% (b) Rotation with 20°

Figure 8: Geometric Transformations adapted to the *Tree* image

Figure 9 shows that the most robust features against scaling were PCA and DCT. They reliably detected copied regions that had been scaled up to 120%. The color based features COLOR 1 and COLOR 2 detected such forgeries up to a scaling of 116% and 114%, respectively. However, it was difficult to distinguish the results of COLOR 2 from the background as the false positive rate was very high at about 0.4. The same problem occured with the features of the MOMENTS 1 method. The false positive rate of DWT was slightly better, but clearly failed to detect anything with scaling more than 106%. Similarly, FMT started degenerating at a scale change of about 106%.

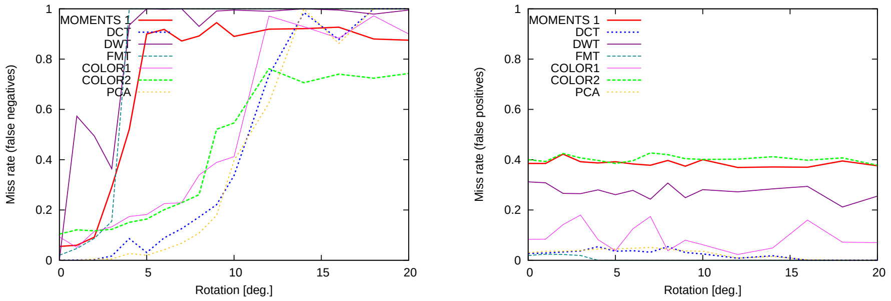
FMT could detect regions rotated up to $3°$ (see Figure 10). This is not surprising, since the FMT features in [BSM09] are only rotation invariant when a brute-force search over all possible feature vector shifts is done, which we left out. Nearly the same result occured for the DWT and MOMENTS 1 features. As with scaling, the color based features gave an average performance. The best features were once again the ones based on PCA and



(a) False negative rate

(b) False positive rate

Figure 9: Scale test – kd-tree sorting



(a) False negative rate

(b) False positive rate

Figure 10: Rotation test – kd-tree sorting

DCT. Note that the results on rotational and scaling invariance are additionally hindered by the use of shift vectors in the verification step. With the shift vectors, a scaled or rotated copied region is decomposed in multiple clusters of similar shift vectors, which flattens the distribution of shift vectors over the image.

Consequently, two problems have to be solved to gain better invariance against geometric transformations. On one hand the feature vector itself has to be robust against geometric transformations and on the other hand a different verification method which is not based on using the same shift vectors has to be established.

5 Conclusions

Copy-move forgery detection is a very active field in image forensics, with a large number of proposed features. We believe that the field has matured enough for a thorough evaluation of the advantages and disadvantages of different approaches. In this work, the features of ten methods were examined on a diverse set of benchmark images. The dataset contains images and copy-moved regions of varying size and texture. We set up a common pipeline for copy-move forgery detection methods and varied intermediate steps, most notably the sorting criterion. We found that, in general, lexicographic sorting yields a lower false positive rate and kd-trees a lower false negative rate. Besides this, lexicographic sorting exhibited severe problems when geometric transformations were applied to the copied region. FMT features showed a very good overall performance. Under geometric transformations, PCA and DCT exhibited remarkably strong results when using same shift vectors as verification criterion.

As part of our future work, we will continue to examine copy-move features under different benchmark conditions. We believe that these findings can be used to create ensembles of copy-move algorithms that are explicitly designed for robustness towards a wide range of postprocessing steps. At the same time, this could help to further reduce the false positive rate, which we identified currently as a major problem in identifying copied regions.

References

[BNO$^+$07] M. K. Bashar, K. Noda, N. Ohnishi, H. Kudo, T. Matsumoto, and Y. Takeuchi. Wavelet-Based Multiresolution Features for Detecting Duplications in Images. In *Conference on Machine Vision Application*, pages 264–267, May 2007.

[BSM09] Sevinc Bayram, Husrev Taha Sencar, and Nasir Memon. An Efficient and Robust Method for Detecting Copy-Move Forgery. *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, pages 1053–1056, April 2009.

[BSN09] S. Bravo-Solorio and A.K. Nandi. Passive Forensic Method for Detecting Duplicated Regions Affected by Reflection, Rotation and Scaling. *17th European Signal Processing Conference*, August 2009.

[Far09a] H. Farid. A Survey of Image Forgery Detection. *Signal Processing Magazine*, 26(2):16–25, March 2009.

[Far09b] Hany Farid. Exposing Digital Forgeries from JPEG Ghosts. *IEEE Transactions on Information Forensics and Security*, 1(4):154–160, 2009.

[FSL03] Jessica Fridrich, David Soukal, and Jan Lukáš. Detection of Copy-Move Forgery in Digital Images. In *Proceedings of Digital Forensic Research Workshop*, August 2003.

[GC08] A. Gallagher and T. Chen. Image Authentication by Detecting Traces of Demosaicing. In *Computer Vision and Pattern Recognition Workshops*, pages 1–8, June 2008.

[HC07] Y. Hsu and S. Chang. Image Splicing Detection using Camera Response Function Consistency and Automatic Segmentation. In *International Conference on Multimedia and Expo*, pages 28–31, July 2007.

[HGZ08] Hailing Huang, Weiqiang Guo, and Yu Zhang. Detection of Copy-Move Forgery in Digital Images Using SIFT Algorithm. In *Computational Intelligence and Industrial Application. Pacific-Asia Workshop on*, volume 2, pages 272–276, December 2008.

[HLWT06] J. He, Z. Lin, L. Wang, and X. Tang. Detecting Doctored JPEG Images Via DCT Coefficient Analysis. In *European Conference on Computer Vision*, volume 3, pages 423–435, May 2006.

[JF05] Micah K. Johnson and Hany Farid. Exposing Digital Forgeries by Detecting Inconsistencies in Lighting. In *Workshop on Multimedia and Security*, pages 1–10, New York, NY, USA, August 2005. ACM.

[JF06] Micah K. Johnson and Hany Farid. Exposing Digital Forgeries through Chromatic Aberration. In *Workshop on Multimedia and Security*, pages 48–55, New York, NY, USA, September 2006. ACM.

[JF07] Micah Johnson and Hany Farid. Exposing Digital Forgeries through Specular Highlights on the Eye. In *International Workshop on Information Hiding*, pages 311–325, 2007.

[KW08] XiaoBing Kang and ShengMin Wei. Identifying Tampered Regions Using Singular Value Decomposition in Digital Image Forensics. In *International Conference on Computer Science and Software Engineering*, volume 3, pages 926–930, December 2008.

[LE07] Jean-François Lalonde and Alexei A. Efros. Using Color Compatibility for Assessing Image Realism. In *IEEE International Conference on Computer Vision*, October 2007.

[LFG06] Jan Lukáš, Jessica Fridrich, and Miroslav Goljan. Digital Camera Identification From Sensor Pattern Noise. *Information Forensics and Security*, 1(2):205–214, June 2006.

[LHQ06] Weiqi Luo, Jiwu Huang, and Guoping Qiu. Robust Detection of Region-Duplication Forgery in Digital Images. In *18th International Conference on Pattern Recognition*, volume 4, pages 746–749, August 2006.

[LWK09] Hwei-Jen Lin, Chun-Wei Wang, and Yang-Ta Kao. Fast Copy-Move Forgery Detection. *WSEAS Transactions on Signal Processing*, 5(5):188–197, 2009.

[LWTS07] Guohui Li, Qiong Wu, Dan Tu, and Shaojie Sun. A Sorted Neighborhood Approach for Detecting Duplicated Regions in Image Forgeries Based on DWT and SVD. In *IEEE International Conference on Multimedia and Expo*, pages 1750–1753, July 2007.

[MS07] Babak Mahdian and Stanislav Saic. Detection of Copy-Move Forgery using a Method Based on Blur Moment Invariants. *Forensic Science International*, 171(2):180–189, December 2007.

[NCLS06] T. Ng, S. Chang, C. Lin, and Q. Sun. Passive-Blind Image Forensics. In *Multimedia Security Technologies for Digital Rights*, chapter 15, pages 383–412. Academic Press, 2006.

[PF04] A.C. Popescu and H. Farid. Exposing Digital Forgeries by Detecting Duplicated Image Regions. Technical Report TR2004-515, Department of Computer Science, Dartmouth College, 2004.

[PF05] A. Popescu and H. Farid. Exposing Digital Forgeries by Detecting Traces of Resampling. *Signal Processing*, 53(2):758–767, February 2005.

[SM08] H. Sencar and N. Memon. Overview of State-of-the-art in Digital Image Forensics. *Algorithms, Architectures and Information Systems Security*, pages 325–344, 2008.

[WLZ$^+$09] Jun-Wen Wang, Guang-Jie Liu, Zhan Zhang, Yue-Wei Dai, and Zhi-Quan Wang. Fast and Robust Forensics for Image Region-Duplication Forgery. *Acta Automatica Sinica*, 2009.